creative computing

Jul-Aug 1978
vol 4, no 4
$2.00

the #1 magazine of computer applications and software

Now Including ROM
COMPUTER APPLICATIONS FOR LIVING

Special Section on Interfacing Your Computer to the Outside World

Business Computing: Word Processing

High-Resolution Graphics Program for the Apple II

Computer Games:
- Backgammon
- Motorcycle Jump

The Rise and Fall of Bar Code

Equipment Reviews:
- Apple II
- Atari Video Computer
- Commodore PET
- Atari Video Pinball

SwTPC MULTI-USER SYSTEM

OPERATES – Up to 4 terminals running INDEPENDENT programs

HARDWARE TIME SHARE – Requires no modifications to computer

IDEAL FOR – All multi-terminal applications

The SwTPC multi-user system converts our standard 6800 single user computer into a multi-user time share system that may be operated with up to four terminals. The four terminals operate independently and may be running four different programs.

No modifications to the computer are necessary, you simply plug in the multi-user board and add an interface for each additional terminal.

The multi-user system is ideal for program training, multi-station business applications and for computer aided instruction (CAI). Speed reduction from a single user system is negligible because all switching is done in hardware.

Multi-user BASIC, suitable for program instruction and simple business applications, is included with the multi-user board. An 8K disc BASIC is also available for systems in which disc drives are used. This software has a complete nine digit floating point math package, full string features and data files.

For computer aided instruction applications, a full feature version of PILOT is available. It includes math operators, misspelling match features and all other proposed for the ANSII standard version. The 6800 multi-user system is just as economical, but far more flexible and powerful than multiple small machines for CAI applications.

MUB-68 Multi-User Board and BASIC Software

Assembled and tested. $150.00

Kit $129.95

SwTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION

219 W. RHAPSODY

SAN ANTONIO, TEXAS 78216

Model Z-2
Up to 512K of RAM/ROM

Model Z-2D
One or two disks
Up to 512K of RAM/ROM
Up to 184K of disk

System Two
Dual disk
Up to 512K of RAM/ROM
Up to 184K of disk

Fill your computer needs with the industry's most professional microcomputers

#1 IN RELIABILITY

When you choose Cromemco you get not only the industry's finest microcomputers but also the industry's widest microcomputer selection.

What's more, you get a computer from the manufacturer that computer dealers rate #1 in product reliability.*

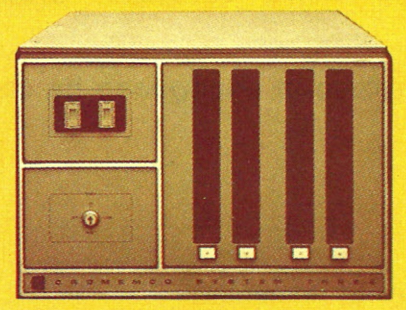
Your range of choice includes our advanced System Three with up to four 8″ disk drives. Or choose from the System Two and Z-2D with 5″ drives. Then for ROM-based work there's the Z2. Each of these computers further offers up to ½ megabyte of RAM (or ROM).

We say these are the industry's most professional microcomputers because they have outstanding features like these:

- **Z-80A microprocessor — operates at 250 nano second cycle time — nearly twice the speed of most others.**

*Rated in ***The 1977 Computer Store Survey*** by Image Resources, Westlake Village, CA.

Up to 512 kilobytes of RAM and 1 megabyte of disk storage



System Three
Two to four disks
Up to 512K of RAM/ROM
Up to 1 megabyte of disk

- **21 card slots to allow for unparalleled system expansion using industry-standard S-100 cards.**
- **S-100 bus — don't overlook how important this is. It has the industry's widest support and Cromemco has professionally implemented it in a fully-shielded design.**
- **Cromemco card support of more than a dozen circuit cards for process control, business systems, and data acquisition including cards for A-D and D-A conversion, for interfacing daisy-wheel or dot-matrix printers, even a card for programming PROMs.**
- **The industry's most professional software support, including FORTRAN IV, 16K Disk-Extended BASIC, Z-80 Macro Assembler, Cromemco Multi-User Operating System — and more coming.**
- **Rugged, professional all-metal construction for rack (or bench or floor cabinet) mounting. Cabinets available.**

FOR TODAY AND TOMORROW

Cromemco computers will meet your needs now and in the future because of their unquestioned technical leadership, professionalism and enormous expandability.

See them today at your dealer. There's no substitute for getting the best.

Cromemco
incorporated
Specialists in computers and peripherals
280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400

CIRCLE 102 ON READER SERVICE CARD

GET THE REAL LOW DOWN BEFORE YOU BUY

SO YOU WANT TO BUY A COMPUTER??

This new book presents an objective look at the top 24 micro systems sold throughout the world.

It discusses the Pros and Cons of each system in No Uncertain Terms and takes a straightforward look at the micro computer industry as it relates to YOU.

Written especially for the layman in a language he can understand. Profit from the mistakes of others.

Includes hundreds of references.

Table of Contents

1. Introduction
2. Don't get hung up on the chips
3. Which category do you fit into
4. Now – About the Hardware
5. Peripherials that plug in?
6. What? No Software!
7. Helpful Suggestions before spending money
8. Addresses

SCIENTIFIC RESEARCH

P.O. Box 490099-B
Key Biscayne, Florida 33149

Phone Orders Call 800-327-6543

WE HONOR master charge THE INTERBANK CARD

BANKAMERICARD

$4.50 per Volume
Volume discounts available

Add $1 handling, all domestic shipments sent U.P.S. except APO and P.O. Box which go parcel post. Foreign orders add $4 for air shipment and make payable in U.S. funds.

in this issue . . .

articles

fiction & foolishness

Jul-Aug 1978 Volume 4, Number 4 Consecutive Issue No. 22

Creative Computing magazine is published bi-monthly by Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. (Editorial office: 51 Dumont Place, Morristown, NJ 07960. Phone: (201) 540-0445.)

Domestic Subscriptions: 12 issues, $15, 24 issues $28, 36 issues $40. Send subscription orders or change of address (P.O. Form 3575) to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. Call 800-631-8112 toll-free (in New Jersey call 201-540-0445) to order a subscription (to be charged only to a bank card).

Second class postage paid at Morristown, New Jersey and at additional mailing offices.

things to do — games

reviews & resources

departments

Foreign Subscriptions:
Great Britain: 12 issues £13, 24 issues £25, 36 issues £36 (surface postage); 12 issues £22, 24 issues £43, 36 issues £63 (airmail postage). Orders to Vincent Coen, 313 Kingston Road, Ilford, Essex, IGI-IPJ, England.
Australia: R. J. Hoess, Electronic Concepts Pty. Ltd., 52-58 Clarence St., Sydney NSW 2000, Australia.
Other Countries: 12 issues $23, 24 issues $44, 36 issues $64 (surface postage, U.S. dollars); 12 issues $39, 24 issues $76, 36 issues $112 (airmail postage, U.S. dollars). Orders to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960, U.S.A.

Publisher
David H. Ahl

Editor-in-Chief
Stephen B. Gray

Managing Editor
Burchenal Green

Associate Editor
Steve North

Technology Editor
Alan Salisbury

Contributing Editors
Frederick W. Chesson
Bill Etra
Louise Etra
Lee Felsenstein
Ed Hershberger
Arthur Karshmer
Theodor Nelson
Robert Osband
Eben Ostby
Stan Veit

Book Service
Barbara Shupe
Debbie Nutt
Joe Ortiz

Retail Marketing
Carol J. Whitaker
Michelle Fisher

Art Director
Patrick J. Gallagher

Marketing Manager
Linda Harrison

Bookkeeper
Jeanne Tick

Systems Analyst
Diana Walters

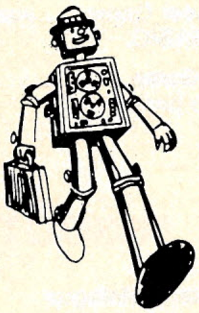
Customer Service
Ethel Fisher
Nancy Hammond

Retail Marketing
Carol J. Whitaker

Subscriptions
Maryann Petrone
Carol Cassata
Sheryl Scalley

Book Service
Barbara Shupe
Debbie Nutt

Order Processing
Ronald Nutt



Advertising Sales

Western States, Texas
Jules E. Thompson
Hearst Building, Suite IIII
5 Third Street
San Francisco, CA 94103
(415) 362-8547

Southern California
Bert Charlton
2560 Via Tejon
Palos Verdes Estates, CA 90274
(213) 378-8361

Mid-Atlantic, Northeast
Charles Lynch
36 Sohier Street
Cohasset, MA 02025
(617) 383-6136

Elsewhere
Burchenal Green
(201) 540-0445

MEMBER
COSMEP
COMMITTEE OF SMALL MAGAZINE EDITORS AND PUBLISHERS

...notices...

Creative Computing Absorbs ROM Magazine

David Ahl, publisher of Creative Computing, announced in early May that *Creative Computing* magazine will absorb *ROM* magazine.

ROM, which has an outstanding editorial staff, was a magazine with a distinctive focus on future imaginative applications for personal computing. "If anything, they were too far ahead of the average personal computer user, many of whom are still struggling to get their systems up and operational," said Ahl. "However, we feel that *ROM*'s articles are a perfect compliment to the how-to material, tutorials software and product reviews of *Creative Computing.*"

ROM will be continued as a separate section of *Creative Computing* and all of the *ROM* columnists and editors will be retained.

ROM subscribers will receive *Creative Computing* for the balance of their subscriptions. Those who subscribe to both magazines will have their subscriptions extended the appropriate number of issues.

Following this consolidation, the paid circulation of *Creative Computing* is over 56,000.

Call For Software

Creative Computing is seeking recreational, educational and personal computing software to be marketed by **Creative Computing Software.**

Specifically, we are seeking software for the following computers:

- Apple II
- Radio Shack TRS-80
- Commodore PET

We are seeking games and other programs which take advantage of the graphics and other capabilities of the system. Programs may be written in BASIC or machine language.

We expect to put between five and fifteen games or educational programs on each tape cassette and sell the cassette for a **retail price of $6.95.** Authors will be paid a **royalty between 10% and 15% of the retail price.** (A number of other companies have been offering higher royalty percentages — up to 50% — and plan to charge more for tapes. Why then, work with **Creative Computing?** Simply because we expect that by offering the customer much more value per dollar than the competition, we will sell significantly more tapes and, therefore, both we and the author will make out better overall.)

We do not want programs that will require any hand-holding or support to the user (no payroll or general-ledger programs please), but we would like things like mailing lists, cataloging, text editing, statistical calculations, etc. On programs of this sort and on very large games, we would expect to market perhaps only one or two programs per tape, possibly at a higher price than $6.95.

Programs should obviously be as bug-free as possible, should be self-documenting and have user aids imbedded in them. We are aiming at a mass market of users where "if anything can go wrong, it will."

Please submit your tape cassette along with any running instructions and 3 first-class (15¢) stamps for return of the cassette to:

Creative Computing Software
P.O. Box 789-M
Morristown, NJ 07960

Please allow 12 to 16 weeks for evaluation and response.

To Game Authors

We'd like to thank you who have submitted games to *Creative* for your contributions, and encourage the rest of you to send us your favorite game or other program. But before you send anything, or even sit down to write a program, please think about the *idea* behind your program. We do not generally accept rehashes of old programs. So if you're thinking of sending us a Biorythm plotter, Blackjack game, or word-puzzle generator, please try something else!

Additionally, once you have a good idea for a program, don't blow it by sending us a poorly written program! You don't have to spend hours and hours developing an optimal sort or search routine—just try to eliminate portions of the program which can never be executed, statements used for debugging, etc. Imaginative and well-written programs are always appreciated here. If you follow these guidelines, you'll increase the likelihood of our accepting your special program and have the satisfaction of seeing your best work in print.

Publication Frequency

You may have noticed on some recent *Creative Computing* subscription notices the price of $15 for 12 issues. This reflects the fact that early in 1979, *Creative Computing* will move to a **monthly frequency.**

Current subscribers will receive the appropriate number of total issues for the balance of their subscriptions. For example, a 3-year subscription at the old $21 rate will receive 18 total issues (bi-monthly through 1978 and monthly thereafter).

The new subscription rates are as follows:

	USA	*Foreign Surface*	*Foreign Air*
1-Year	$ 15	$ 23	$ 39
2-Year	$ 28	$ 44	$ 52
3-Year	$ 40	$ 56	$ 64
Lifetime	$300	$400	$600

PLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR

Huge Discounts!

HOBBY WORLD ELECTRONICS

Savings up to 70% on major brand IC parts and computer kits. For complete IC listings write for our catalog.

KITS

WAMECO	S-100 P.C. Boards	
	8K RAM	28.00
	8080 CPU	28.00
	12-Slot Mother Board	33.00
ITHACA AUDIO	S100 P.C. Boards	
	8K RAM	28.00
	Z-80 CPU	28.00
	2708/2716 EPROM Boards (New)	28.00
	W/W Prototype Board (New)	25.00
SOLID STATE MUSIC	S-100 Kits & Bare Boards	
MB6A 8K	Kit	129.95
STATIC RAM	Bare Board	25.95
MB7 16K STATIC RAM		
Kit		435.00
Bare Board		25.95
MB8 8K/16 K EPROM USES 2708's		
Kit Less EPROMs		75.95

SPECIAL

HUH ELECTRONICS 6 ten minute data cassettes: $6.50 (New)

5¼" MINIDISCETTES: Soft Sector, 10, 16, each $4.65 (New)

8" FLEXIBLE DISCS: Hard or Soft Sector, each $5.33 (New)

KITS

SOLID STATE MUSIC	
VB1B VIDEO INTERFACE	
Kit	129.95
Bare Board	25.95
RF Modulator TV-1 Converts Video Output to RF For TV antenna terminals, complete Kit	8.95
NEW FROM CYBERCOM DIV OF SSM	
MB8A—2708 16K EPROM with Vector Jump Kit	85.00
OB-1 Vector Jump & Prototype	
Kit	49.00
Board	25.95
104 2+2 I/O Kit	139.95
SYNTHESIZER SB-1 MUSIC Kit with Software	145.00
MT-1 15-Slot Mother Board	39.95
XB-1 EXTENDER BOARD Bare Board	8.99
SSM 8080 MONITOR VI	
ON 2-2708	47.00
ON 8-1702A	47.00

BOOKS

The Basic Workbook	$ 5.50
Programming Proverbs	6.95
Discovering BASIC	6.85
COBOL with Style	6.95
Advanced BASIC	7.95
Standard Dictionary of Computers & Information Processing	16.95
Game Playing with Computers	16.95
Game Playing with BASIC	6.95
Minicomputers	13.95
Microprocessors	10.95
Digital Experiments	8.95
Digital Signal Analysis	19.95
Digital Troubleshooting	9.95
110 CMOS Digital IC Projects	5.95
400 Ideas for Design, Volume 2	13.75
Analysis and Design of Digital Circuits and Computer Systems	22.50
Telephone Accessories You Can Build	4.95
Basic Electronic Switching for Telephone Systems	6.95
Basic Carrier Telephony	6.85

(Continued)

8080 Machine Languages Programming	6.96
Robots At Your Doorstep	6.95

BEGINNERS BOOKS

Beginning BASIC	9.95
Home Computer Beginners Glossary and Guide	6.95
Basic BASIC	$ 8.95
Introduction to BASIC	8.95
Home Computers: 210 Questions and Answers Volume 2: Software	6.95
Microcomputer Dictionary and Guide	19.95
Microprocessor Basics	10.95
Home Computers: 210 Questions and Answers Volume 1: Hardware	7.95
Understanding Integrated Circuits	4.95
Semiconductor Circuit Elements	6.95
Fundamentals and Applications of Digital Logic Circuits	7.95

74L$00 TTL

74L$00	.21	74L$138	.70
74L$02	.21	74L$139	.70
74L$04	.24	74L$151	.65
74L$08	.21	74L$153	.66
74L$10	.21	74L$154	1.00
74L$14	.85	74L$157	.62
74L$20	.23	74L$160	.82
74L$21	.23	74L$161	.82
74L$22	.23	74L$162	.82
74L$30	.23	74L$163	.82
74L$32	.30	74L$164	.98
74L$37	.31	74L$174	.75
74L$38	.31	74L$175	.79
74L$42	.60	74L$190	.90
74L$47	.75	74L$191	.90
74L$48	.72	74L$192	.90
74L$73	.35	74L$196	.80
74L$74	.35	74L$197	.80
74L$75	.53	74L$221	1.06
74L$76	.37	74L$257	.71
74L$86	.36	74L$258	.70
74L$90	.52	74L$266	.26
74L$92	.52	74L$283	.72
74L$93	.52	74L$365	.55
74L$109	.36	74L$366	.55
74L$112	.36	74L$367	.55
74L$113	.36	74L$368	.55
74L$114	.36	74L$386	.39
74L$125	.46	81L$95	.77
74L$126	.46	81L$96	.77
74L$132	.75	81L$97	.77
		81L$98	.77

74XX

7400	.11	7482	.50
7401	.13	7483	.62
7402	.13	7485	.75
7403	.13	7486	.26
7404	.15	7489	1.75
7405	.13	7490	.40
7406	.16	7491	.51
7407	.23	7492	.40
7408	.17	7493	.40
7409	.17	7494	.60
7410	.13	7495	.60
7411	.18	7496	.60
7412	.13	74107	.28
7413	.25	74109	.22
7414	.61	74121	.29
7416	.24	74122	.38
7417	.22	74123	.48
7420	.13	74132	.65
7423	.25	74141	.70
7425	.25	74145	.65
7426	.22	74150	.88
7427	.17	74151	.61
7430	.13	74153	.61
7432	.23	74154	.95
7437	.21	74157	.55
7438	.21	74161	.55
7439	.25	74163	.55
7440	.13	74164	.85
7441	.70	74165	.90
7442	.37	74170	1.69
7443	.59	74173	1.10
7444	.59	74174	.85
7445	.65	74175	.75
7446	.62	74176	.69
7447	.59	74177	.70
7448	.60	74180	.65
7450	.13	74181	1.75
7451	.13	74190	.95
7453	.13	74191	.95
7454	.13	74192	.79
7460	.14	74193	.80
7470	.26	74195	.49
7472	.21	74221	.86
7473	.21	74251	1.00
7474	.27	74365	.62
7475	.45	74366	.62
7476	.28	74367	.62
7480	.31	74368	.62
7481	.95		

SPECIAL

(New) 25 Pin—D Subminiature Solder Tail
Male 3.00
Female 3.50

100 PIN EDGE CONNECTOR, Imsai spacing 3.50 (New)

SUPPORT DEVICES

6820	8.00
6850	8.00
8212	3.45
8214	8.00
8216	3.75
8224	3.50
8228	6.25
8226	3.85
8238	7.95

MEMORY

1702A	3.75
2708	11.50
2716 TI	22.50
21L02 450ns	1.25
21L02 250ns	1.60
2114	8.50

MICROPROCESSOR

8080A	11.50
Z-80	24.95
Z-80A	34.95
6800	16.50

CMOS

4001	.16	4025	.16
4002	.16	4028	.74
4006	.99	4030	.22
4007	.16	4035	.98
4008	.79	4040	.99
4009	.37	4049	.34
4010	.37	4050	.34
4011	.16	4051	.89
4012	.16	4052	1.15
4013	.29	4053	.89
4014	.74	4066	.59
4015	.74	4071	.19
4016	.29	4081	.23
4017	.79	4082	.23
4018	.79	4507	.39
4019	.23	4510	1.05
4020	.84		
4021	.89		
4022	.89		
4023	.16		
4024	.67		

HOBBY WORLD

5716 W. Manchester Ave.
Suite #5
Los Angeles, CA 90045

TELEPHONE ORDERS:
Call (213) 641-4200

master charge THE INTERBANK CARD

BANKAMERICARD VISA

☐ Cash
☐ COD
Charge My
☐ M.C.
☐ BAC (VISA)

Exp. Date: ______
\# ____ — ____ — ____
Sig. ______

☐ Send your complete catalog, quickly.
☐ Please send me the following items I have listed below:

Qty.	Stock No.	Price
	Postage/Handling	$1.50

Name ______

Address ______

City ______ State ______ Zip ______

Satisfaction 100% Guaranteed

California Residents Add 6% Sales Tax

Note: Minimum Order $10.00, 5% Discount over $100.00 on I.C.'s only.

NEW 1978
IC MASTER

- Over 40,000 IC's listed.
- Over 2,000 pages.
- Updated every 90 days.

Retail Value **$55.00**

Your Price **$39.99**

Check catalogue for more listings on linears, voltage regulators, resistor books, IC sockets and more.

IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE YOUR COMPLETE SATISFACTION IS OUR GUARANTEE

Service and Support

No matter how marvelous a machine a manufacturer's computer is, none of the ingenious circuitry will matter much if the service and support is inadequate. I had the opportunity to observe this at first hand in my first job in data processing, as an IBM field-service engineer, or Customer Engineer, as IBM puts it. When an IBM machine stopped working, you usually got service the same day. Most of the competition would provide service on their machines *maybe* the same day, usually the next, often the day after that. If you needed help with operating the machines, writing programs, designing forms, or just about anything at all, IBM provided as much help as you needed, for as long as you needed it. That's what made IBM the giant it is. I saw several large installations of competing computers taken out, and IBM computers put in, simply because there was just too much "down time" on the other machines.

Personal computers used in business, and business computers made by personal-computer manufacturers, will very soon face the same problem, of requiring adequate service and support. I've heard of an entrepreneur in California who has over a dozen microcomputers installed in businesses, none more than 30 miles from his office. His mechanic is a board-switcher, who doesn't carry a "loaner" in his car. What if a businessman has to meet a payroll this afternoon, and his Dinky-80 has a faulty front-panel switch that not all the board-switching in the world will cure? Unless the entrepreneur has a similar machine on the shelf at the office, and can get it to the businessman in the next hour or two, the payroll checks will have to be made out by hand. How many times can this or similar problems occur before the businessman begins to think he's bought a lemon?

The current thinking of the computer people at Heath is that they don't want to support business uses of their H-8 and H-11 computers. What businessman could wait for as long as it takes for all or part of his H-11 to get sent to Benton Harbor, get repaired, and returned? The only solution for Heath would be to set up service and support offices around the country, providing not only repair service but programming help. Many businesses may wish to use their micros on two shifts, which would require extra field-service and systems-analyst people. What do you think such a field-office network would cost Heath to maintain? What is the salary of a really good microcomputer repairman? How many such men are there in the United States right now?

Unfortunately for the manufacturers of personal computers, there is no easy answer to the problem of service and support. How many businessmen would be willing to buy a backup computer? If a businessman is buying a personal micro because he can't afford a commercial mini, how could he possibly afford to have a serviceman permanently on the premises?

But aren't there national service organizations that repair computers all over the country? Yes, there are several, including Western Union, ITT and Honeywell. One microcomputer manufacturer told me he went to one of these and got their facts and figures. He would have to train the service company's maintenance people how to repair his computer, and provide parts for each one of their offices in the USA. The cost of replacement boards alone would be $380,000. Another such company said they could offer a cheaper but slower service: a user sends his computer to the service company by air express, gets a replacement *within three days.* This manufacturer is thinking of having dealers provide service contracts for $100 a month for total maintenance, assuming the businessman is within 30 miles of the dealer; beyond 30 miles, add $20 an hour travel time and 20 cents a mile. What if you're 50 or 60 miles from the dealer, and the machine breaks down several times a month?

It is entirely possible that within the next year or two, there will be enough disillusioned businessmen who have been burned by low-price microcomputers with too-high down-times, to spread the word that if you want to avoid all those micro-headaches, go to IBM, to DEC, to Data General, and the other companies that can afford to provide the kind of service and support a businessman needs. I've spoken to several micro-makers about this problem, and not one has a ready solution. Do you?

--Stephen B. Gray

Interfacing to the Outside World

Once you've learned to program your microcomputer, and have tried out a couple of the games and CAI programs from the pages of *Creative Computing*, you may be looking for new worlds to conquer. How about connecting your machine to the outside world? We've gathered four articles in this issue to show various ways of making such connections.

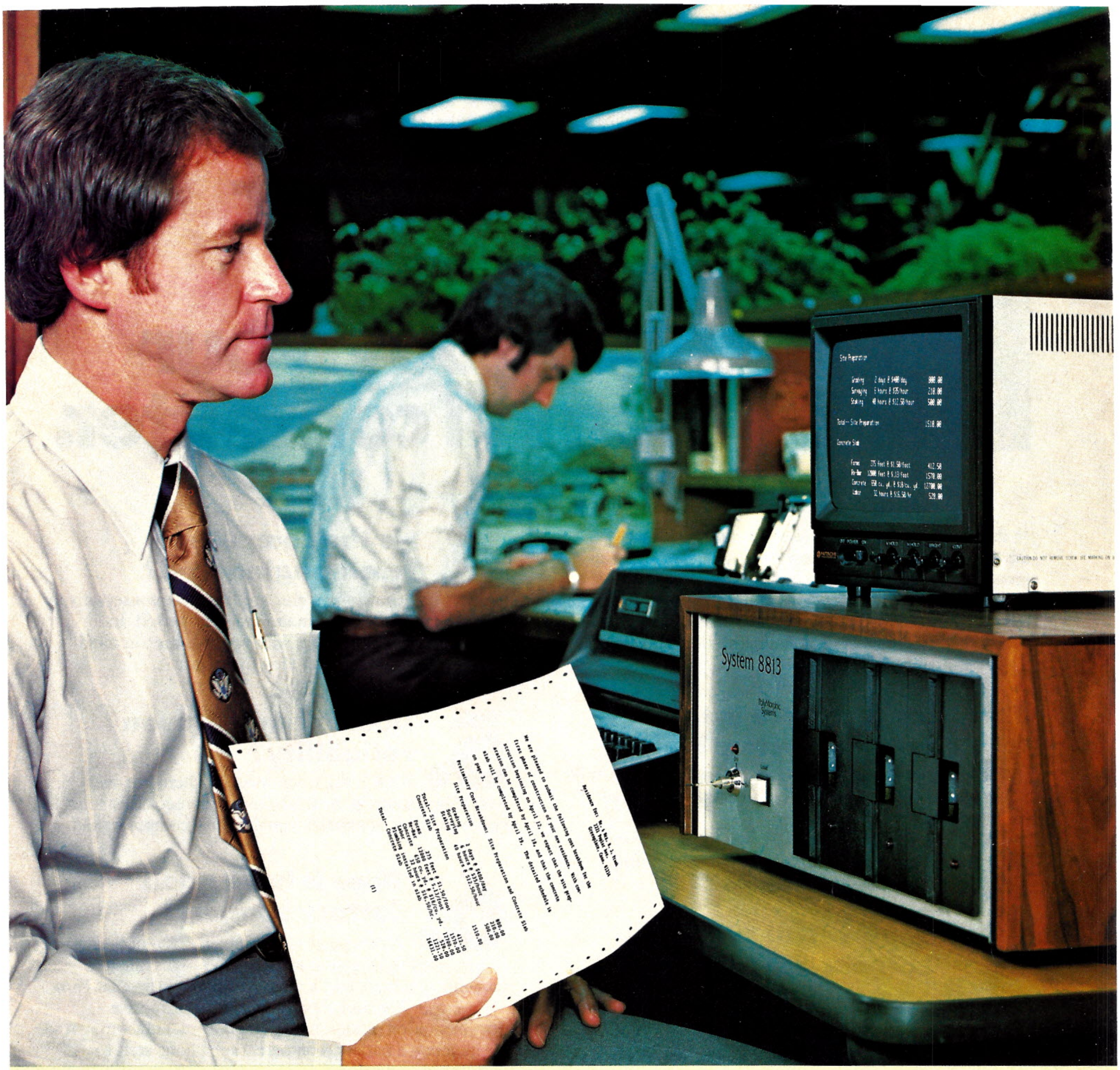
The Bit Pad converts graphic information to digital form, for input to a computer. Or you can control appliances and other electrical devices anywhere in your home or office building, either through the AC power lines with the Mountain Hardware system, or over individual wires, using a different variety of interfacing hardware, with the Digital Group system.

You may never develop a system as complex as the one Cromemco did to process EKG readings, but the article does show just how complicated it can be to convert analog signals to digital and process them. Chances are even less that you'll ever build anything as complicated as described in "Man-Machine Tactile Communication," but this does show just how far you'd have to go to "feel and identify shapes and objects existing only in the memory of a computer."

The applications described in these articles are only the beginning of a world in which microcomputers may become almost as commonplace as telephones, in guiding the operations of dozens of electrical and electronic devices that we ourselves interface with daily.

Notice to ROM Subscribers

Until the **ROM** and **Creative Computing** subscription lists get merged, you may receive two copies of **Creative Computing.** Don't worry! If you have subscribed to both magazines, your subscription to **Creative** will be extended the appropriate number of issues. We expect to have this all straightened out by the Sep/Oct issue.



The Computer for the Professional

The 8813 was built with you, the professional, in mind. It quickly and easily processes cost estimates, payrolls, accounts, inventory, patient/client records and much more. You can write reports, briefs, and proposals on the 8813's typewriter keyboard, see them on the video screen, and instantly correct, revise, or print them.

Using the 8813, one person can process what would normally require many secretaries, several bookkeepers, and a great deal of *time.* And data storage takes a small fraction of the *space* used by previous methods.

You don't need to learn complicated computer languages. The 8813 understands commands in English. If you want to write your own programs, the 8813 includes a simple computer language, BASIC, that you can master in a few days. The 8813 slashes the professional's overhead. It's a powerful time and money-saving ally. Prices for complete systems including printer start at less than $8,000.

See the 8813 at your local dealer or contact PolyMorphic Systems, 460 Ward Drive, Santa Barbara, California, 93111, (805) 967-0468, for the name of the dealer nearest you.

PolyMorphic Systems

Seven points to consider before you buy your small computer.

In this magazine, alone, there are probably a dozen ads for small computers. New companies are breaking ground like spring flowers.

How, then, do you determine which computer offers the features you need most... at the price you can afford?

We'd like to propose seven basic questions to help you make an intelligent decision.

1. How complete is the computer system?

Many buyers of small computers are in for a rude awakening when they have to spend additional money for interfaces.

The Sol-20 Terminal Computer was the first *complete* small computer system. Everything you need to make it work is included in the basic package.

2. Is powerful system software available?

It won't do if your system is "tongue-tied."

Processor Technology Corporation has devoted more effort to the development of software than any other small computer maker. Our latest offering is the first fully implemented disk operating system for a small computer: PTDOS. It contains over 40 major commands, several languages and numerous utilities. Our high level languages include Extended BASIC, Assembler, FORTRAN,* FOCAL and PILOT.*

3. Is the system easy to expand?

More and more computer owners are expanding their small computers to handle business and other specialized requirements.

The largest Sol system can handle 64K bytes of RAM memory and operate with a three megabyte on-line disk memory. Sol systems use the S-100 Bus. So you can use a wide variety of hardware.

*Available soon.

4. Is the computer well-engineered?

Our Sol systems are the most conservatively rated and ruggedly built in the industry, period. In addition we designed them with you, the user, in mind; Sols are easy to build and a joy to operate.

5. Does it have proven reliability?

What is the track record? There are over 5,000 Sol systems in the field. Our track record for reliable performance is unparalleled in the small computer field.

6. Does it have good factory support?

A computer is a complex piece of hardware. So you want to be sure it is backed up with complete manuals, drawings and a factory support team that cares.

Processor Technology offers the most extensive documentation of any small computer manufacturer. And we maintain a patient, competent telephone staff to answer your questions.

7. Are maintenance and service people accessible?

Where are they located?

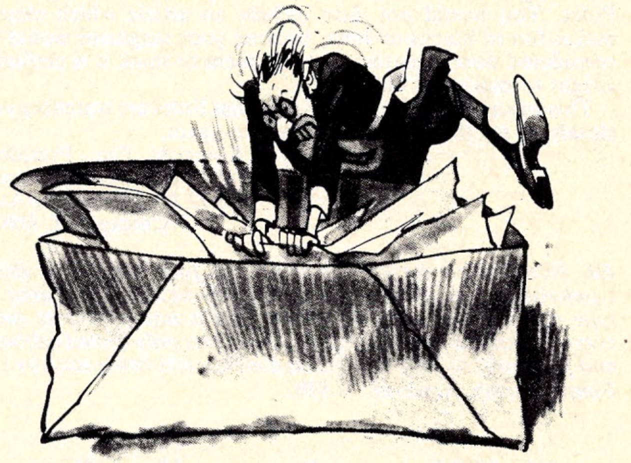
Processor Technology has maintenance and service people in over 50 cities around the U.S.

As you continue turning the pages, see how we stack up to the other computers in this magazine. If we've succeeded in whetting your appetite, see your Sol dealer or write for information on the complete family of Sol computers.

Processor Technology Corporation, Box C, 7100 Johnson Industrial Drive, Pleasanton, CA 94566. (415) 829-2600.

Processor Technology

input/output



Kudos for Datamazing

Dear Editor:

I liked your issue of *Datamazing* (Mar-Apr 1978) but I must take a little umbrage* with you.

I think you cropped too much of Linda Eckerstrom; she's definitely worth seeing more of. Phillip Tubb's poem would have worked much better as a haiku. And a bit of erroneous information: the Cray-1 will not be manufactured on a chip. The next machine from the wizard of Chippewa Falls' laboratory will be the Cray-2. Only one machine will be made. Cray himself will be the cpu.

Otherwise I think it was one of the best parodies of Infosystems I've seen. Keep up the good work.

John L. Kirkley, Editor
Datamation
1801 S. LaCienega Blvd.
Los Angeles, CA 90035

*Umbrage is best taken with a group of close friends. A little Bach on the hi-fi and a respectable California white wine makes it all the better.

First the LED, now the SED

Dear Editor:

In your April 1978 issue of Datamazing, you didn't mention HITWEC's (Hole-in-the-Wall-Electronics-Corp.) new Silicon or Sulfur Smell-Emitting Diodes.

The basic purpose of Smell-Emitting Diodes is to apply 117 V AC and get large-scale disintegration.

The beauty of using the Silicon or Sulfur technique is that you get increased packing density, faster destruction time, and a higher smell-to-watt ratio.

Readers needing more information should write Gross Ripov, President, HITWEC, 2900 Germanium Road, Sulfur Springs, Texas.

Rob Cove
1711 Plymouth Dr.
Irving, TX 75061

put...input/output...in

"Know That Poles Are Watching You"

Dear Editor:

This letter is in regard to an article in your magazine Vol. 4, No. 2, pg. 69, which was an outstanding case of defamation of Poles. You would not dare include an article which clearly makes fun of Blacks or Jews, because your magazine would be considered discriminatory, yet you seem to think it is perfectly alright to make fun of Poles!

This was a terrible blunder. Know that Poles are watching you closely to see that it does not happen again.

Thaddeus L. Kowalski, Esq., President
Polish American Congress
2952 N. Milwaukee Ave.
Chicago, IL 60618

Ed. Note: Upon further checking, we found that the diesel typewriter described in the Datamazing section is not *used to type menus for the Polish Army. This is not surprising since everyone knows that the Polish Army eats only boiled cabbage and potatoes, so they don't* need *a menu. It is being used by the Finnish Army, however. —SBG*

Shell-Metzner Sort vs Hart Sort

Dear Editor:

As sorting is an operation frequently used on our computer (8K,PDP11/10) I was interested in the article by Richard Hart (Jan-Feb 1978, pp. 96-101). At present we are using the Shell-Metzner sort which was described by John P. Grillo (Nov-Dec 1976, pp. 76-80).

This latter sort requiring only about a quarter of the statements required by method described by Hart. On comparing sorts within the limitations of our BASIC—maximum array size 255—I find the Shell-Metzner sort is also faster in sorting randomly generated numbers.

Number sorted	*Shell-Metzner (Time in seconds)*	*Method described by Hart*
50	21	27
100	48	61
200	121	135

Within these values the Shell-Metzner wins on two counts (1) compactness of the algorithm and (2) faster sorting.

Perhaps someone who has a computer that can handle larger arrays could look at comparisons above 200.

Pat Fitzgerald
Winchmore Irrigation Research Station
Ministry of Agriculture & Fisheries
Ashburton, New Zealand

Structured Programming

Dear Editor:

I thumbed through your Mar-Apr. 1978 issue and came across Alan B. Salisburg's article on structured programming (pp. 58-64). Quite useful as a primer on the subject, and on aspects of implementing "goto-less" programming in a non-structured language, but I wonder about the applications to home computers. One must remember that most home computer BASICs are interpreters, and all this structured stuff will increase execution time and memory used. Structured languages (PL/I, PASCAL, Algol) can get away with it because they are almost always implemented as compilers on *big, fast* machines, and these compilers can afford to optimize the object code they produce. In other words, you can tell a CPU to jump very easily, but to explain to it that this variable here is local and that one over there is global, well, that's another matter.

Thomas Wood
248 Waterford Road
Oakdale, NY 11769

Music Boards

Dear Editor:

Thank you for Steve North's article "Computer Music: Four Systems Plus a Music Program" (Mar-Apr 1978 *Creative*). It was a well-researched, fair and unbiased article and a valuable guide to consumers interested in getting music from their computers.

We would like, however, to note the following:

(1) Both the Model 6 Newtech music board (for the S-100 bus) and Model 68 (for the SWTPC 6800 computer) do *not* need an external amplifier. The on-board audio amplifier can easily power a small, efficient 8-ohm speaker if the user wants more volume than the two-inch on-board speaker can provide.

(2) As noted in the article, the Newtech boards can produce two-part music. It should be pointed out that each "voice" can have the same or different sound qualities, like the different "stops" on an organ; the music can be output as mono or stereo (using two boards). The two-part software will be released shortly.

(3) Perhaps because its focus was on boards that produce *music*, the article failed to mention that the Newtech boards can produce a wide range of other sounds as well — for example, Morse code, touch-tone synthesis, and white noise, "phasor" and telephone-ringing sounds, suitable for use with computer games.

Since we didn't point out these features to Steve North when he was researching the article, we don't mean in any way to detract from the overall excellence of the article.

Dorothy Siegel
Vice President
Newtech Computer Systems, Inc.
131 Joralemon St.
Brooklyn, NY 11201

Steve North adds: It also occurred to me that the technique used for locating program lines in my program for the Newtech Music Board is horribly inefficient. Use of a binary search or some other search designed to optimize search time, when inserting consecutive lines between existing lines, would speed things up a bit.

Racetrack Program: Missing Digits

Dear Editor:

My program, Racetrack, was published in the March-April 1978 *Creative Computing*. It has come to my attention that the last line reference was ommitted in lines 1240 and 2010. I have received much mail regarding this ommission. I would appreciate publication of the corrected lines that follow.

Line 1240 should read:

```
ON G1(I*I+2) GOTO 01250,
01460,01310,01340,01480
01400,01370,01280,01430
```

Line 2010 should read:

```
ON G1(I*I+2) GOTO 02900,
02020, 02900, 02020, 02020,
02020, 02900, 02020, 02900
```

Scott Bennett
709 Diana Court
Iowa City, IA 52240

Sysgenesis

Dear Editor:

I noticed, in your "Software Spotlight" in the Datamazing parody section of the March-April 1978 issue (p 78), an item titled "User Documentation, Region Earth." This is something I wrote for the magazine put out by the New York University science-fiction society, NYUSFS, about three years ago. I was happy to see it in your magazines, although if any credit is due, I'd like some. Also, I noticed a few errors and omissions, and will be glad to send a copy of the original to anyone who would like it, and who will also send a SASE.

Robert Katz
136 Burns Ave.
Lodi, NJ 07644

The New MSI System 12

The MSI System 12 computer system combines the popular MSI 6800 processor...complete with 32K of memory...the MSI FD-8 QUAD floppy disk system, and the new MSI HD-8/R 10 megabyte fixed/removable hard disk system in one compact desk unit.

Ideal for business applications, the MSI System 12 gives you a large capacity hard disk for mass storage, and a floppy disk system for program loading, back-up, software updates and exchanges. The new SDOS operating system is employed to integrate the two disk systems together.

Complete with CRT, high speed printer, and convenient desk unit, the MSI System 12 is one of the most powerful micro-computer systems available today.

Arizona

Personal Computer Place
1840 W. Southern
Mesa, Arizona 85202

California

A-Vidd Electronics
2210 Bellflower Blvd.
Long Beach, California 90815

Computerland of San Mateo
42 W. 42nd Ave.
San Mateo, California 94403

Florida

Microcomputer Systems
144 S. Dale Mabry Highway
Tampa, Florida 33609

Illinois

American Microprocessors
Equipment & Supply Corporation
20 N. Milwaukie
Half Day, Illinois 60069

Computerland of Arlington Heights
50 E. Rand Road
Arlington Heights, Illinois

4C Corporation
P.O. Box 530
Mundelein, Illinois 60060

Lillipute Computer Mart, Inc.
4446 Oakton St.
Skokie, Illinois 60076

Midland Standard
P.O. Box 38
603 E. Chicago St.
Elgin, Illinois

Wysocki Electric
6563 11th St.
Rockford, Illinois 61109

Iowa

Electronic Data
1200 Locust
Des Moines, Iowa 50301

Kansas

Barney & Associates
Electronics Division
425 N. Broadway
Pittsburg, Kansas 66762

CMPTR-C
704 Taylor
Topeka, Kansas 66604

Louisiana

Freeman Electronics
1100 Ridge Ave.
West Monroe, Louisiana 71291

Gallion Data Systems
3103 Malvern
Bossier City, Louisiana 71111

Maryland

Computer Workshop
1776 E. Jefferson
Rockville, Maryland 20852

Missouri

Gallion Data Systems
201 N. 11th St.
Blue Springs, Missouri 64015

H & K Systems
15 E. 31st St.
Kansas City, Missouri 64108

Montana

Compact Computers
113 Hamilton
Butte, Montana 59701

New York

Computer Mart of New York
118 Madison Avenue
New York, New York 10006

Oklahoma

High Technology
1020 W. Wilshire Blvd.
Oklahoma City, Oklahoma 73116

Pennsylvania

The Electronics Place
7250 McKnight Rd.
Pittsburgh, Pennsylvania 15237

Gallion Data Systems
908 Knepper Drive
Mechanicsburg, Pennsylvania 17055

Texas

The Computer Shop
6812 San Pedro
San Antonio, Texas 78216

Washington

Digitek
5950 Sixth Avenue South
Suite 101
Seattle, Washington 98108

Midwest Scientific International
Chausee De Charleroi, 80
1060 Brussels, Belgium
Telex 26025

Canada

First Canadian Computer Store
44 Eglinton Avenue West
Toronto, Ontario M4R 1A1
Canada

England

Strumech Engineering
Electronics Division
Portland House
Coppice Side, Brownhills
Walsall Staffordshire
England

Computer Workshop
174 Ifield Rd.
London, England SW109AG

Germany and Austria

C.O.I. Systeme Munchen
EDV Vertriebsgesellschaft mbH
Arabellastraße 5
8000 Munchen 81
Germany

Belgium

Computer Resources
Chausee De Charleroi, 80
1060 Brussels, Belgium

Switzerland

Agence De Distribution et Vente
Case Postale 801
1211 Geneve 1
Switzerland

Australia

Sontron Instruments
Byte Shope
17arawatta St. Carnegie
Victoria, Australia 3163

The Netherlands

MRL Ectronics
Postbus 88-Delft, Foulkeslaan 100
The Netherlands

South Africa

Radiokom
Cnr. George St. & Hendrik Verwoerd Dr.
Randburg, Transvaal
South Africa

Venezuela

Tramboca (Sistema Pek 2000)
Centro Peru, PISO 2-Ofic. 23
Caracas, Venezuela

Midwest Scientific Instruments

220 W. Cedar Olathe, Kansas 66061 (913) 764-3273
TWX 910 749 6403 (MSI OLAT) TELEX 42525 (MSI A OLAT)

CIRCLE 108 ON READER SERVICE CARD

put...input/output...in

360/370 Emulator Chip

Dear Editor:

Regarding Victor Kay's letter about bridge asking if Bridge can now be programmed on a home-size computer (Nov-Dec 1977, p 12):

I did all of his requests many years ago. It would be easy to put on a micro. I would run directly if we had an IBM 360/370 emulator chip. (Wow!, think of the application software available if we had that!)

Which leads to two important questions. Why not print *flowcharts* and documentation rather than listings? I can't (won't!) run BASIC. You most probably can't run my languages (APL, Fortran, ASM).

Secondly, why should anyone *give away* programs for nothing? Trade, yes; sell *reasonably*, yes' give away, no! Does someone care to write up a careful analysis of the economics of micros and their software?

William B. Adams
P.O. Box 34683
Bethesda, MD 20034

Steve North replies: Although several people say you can make BIG bucks selling software, it has yet to happen. Publishing flowcharts is rather impractical; they are bulky and are felt by some to be outdated. I wouldn't be surprised to see a 370 emulator chip in a couple of years. BUT, IBM isn't going to give away its software. As Alan Salisbury pointed ou, you could emulate a 370 right now using bit-slices. Remember, a bit-slice is a little piece of a Floppy ROM!

Wanted: Computer Misteaks

Freelance writer is seeking amusing anecdotes of mistakes attributed to computers, for a magazine article. He already has a number of them including the magazine subscriber who received 700 copies of his favorite magazine and the department store customer who received bills for $ millions. Any funny, or not so funny, story will be appreciated. Credit will be given if requested. Send to Wm. R. Gerler, 1315 Valley View Drive, Racine, Wisc. 53405

Reply to "IBM Hater"

Dear Editor:

Unfortunately, "IBM Hater" (p 10, Jan-Feb 1978) missed the entire point of my letter. What is unique about the IBM 1130 is the range and quality of *software* available for it such as the many compilers (I listed 12) and an operating system so easy to use that it makes the telephone seem complicated.

I also never suggested that anyone buy an IBM 1130, since it was designed about 15 years ago and of course there are newer machines that are both cheaper and faster. In fact, my letter was a response to a previous letter that put down the 1130.

However, I shall wait for Mr. or Ms. Hater to name an actual machine which timeshares all 12 compilers, at the same time; with a simply control language and which will provide a reasonable response time for 25 users. I pick 25 because that's how many people use our computer at the same time. Also this machine should be as reliable as the IBM 1130 with the same quality of repair service and software documentation as is available (I will agree, at too high a price) from IBM.

I will be waiting for a long time.

Jim Berlin
167-01 Gothic Drive
Jamaica, NY 11432

Butterfly-Sort Comparison

Dear Editor:

Thought you might be interested in the enclosed item, taken from Richard Lerseth's "Commentary" column in the February 1978 *Push & Pop Newsletter* of the Sacramento Microcomputer Users Group (SMUG).

Dennis Church
Sacramento, CA

NOTE ON SORTING: I implemented in FORTRAN the butterfly sort written up in CREATIVE COMPUTING, January-February 1978 issue. Dallas Parcher also has done the same in MITS Basic (I am not sure which version). Believe me, it works. Here are some statistics:

	No. of	seconds for Sort on CDC-3300					
No. of Random No.	10	100	500	1000	2000	2000 Sorted	2000 Reverse Sorted
Bubble Sort	.005	.388	9.8	38.3	157.8	0.1	192.2
Butterfly Sort	.009	.095	0.5	1.1	2.3	1.5	1.5

So, if this tells you anything, the bubble sort is a terrible sort at the very least. Hop to it you sort fans.

Ed. note: Although author Richard Hart called it a "butterfly merge," he gave no name to the sort itself. When contacted, he said, "I'm sure the sort will name itself; Hart sort or butterfly sort is fine with me."

A Gr&st& Play?

Dear Editor:

Perhaps you would consider the following true story for *Creative Computing:*

I spent several weeks typing up names and addresses from a mailing list onto special computer-input forms, which were to be used for printing computer-prepared mailing labels.

When the Data Processing Supervisor noticed that I had written out the word "and" in each Mr. *and* Mrs. name on the list, he decided to shorten the computer processing time by issuing instructions to the computer to substitute the ampersand "&"wherever the letters "and" appeared.

The mailing labels came out fine, except that all the people named "Andrew" were addressed "&rew" ... and all surnames such as "Anderson" came out reading "&erson."

They are still trying to straighten out the mess!

Susan Kramer
156 Grand Boulevard
Massapequa Park, N.Y. 11762

Simulation Program Error

Dear Editor:

I enjoy *Creative Computing* magazine very much and was especially interested in the article on the computer simulation of electric power generation (Jan-Feb 1978, p 88) as this is related to some of the work that I do.

However when I looked at the results of the simulation as presented in the magazine, they did not make sense based on other studies that I have run. Upon examining the program listing I found what appears to be an error in assigning non-electrical fuel consumption.

Specifically as follows:

1. One-year rather than five-year consumption is subtracted from the reserves (lines: 430, 660).
2. After the first time period the rate of non-electrical demand is never increased (line 1640 goes to line 500 rather than line 420).

This should produce a substantial and realistic change in the results of the simulation, illustrating again the value of knowing what kind of results to expect when checking out a computer program.

David E. White
85 Jason Street
Arlington, MA 02174

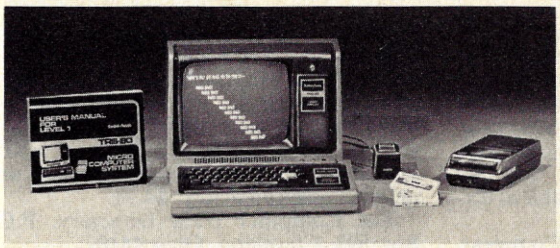
Thank you very much for pointing out this error. Your observation on knowing what to expect is certainly valid (I didn't know) but Murphy's (or somebody's) law is also operative that says that any program over 50 lines long has a bug in it after it has been completely debugged.—DHA

Radio Shack's personal computer system?
This ad just might make you a believer.

You can't beat the 4K system at $599

TRS-80 "Breakthru"
- TRS-80 microcomputer
- 12″ video display
- Professional keyboard
- Power supply
- Cassette tape recorder
- 4K RAM, Level-I BASIC
- 232-page manual
- 2 game cassettes

...or the step-up 16K system at $899



TRS-80 "Sweet 16"
- Above, except includes 16K RAM

...or the fast 4K/printer system at $1198

TRS-80 "Educator"
- Above, except includes 4K RAM and screen printer

...or the Level-II 16K/printer/disk system at $2385

TRS-80 "Professional"
- Above, except includes 16K RAM, disk drive, expansion interface, and Level-II BASIC

So how are you gonna beat the system that does this much for this little? No way!

...The amazing new 32K/Level-II/2-disk/ line printer system at $3874

TRS-80 "Business"
- Above, except includes 32K RAM, line printer, and two disk drives

Get details and order now at Radio Shack stores and dealers in the USA, Canada, UK, Australia, Belgium, Holland, France, Japan.
Write Radio Shack, Division of Tandy Corporation, Dept. C-042, 1400 One Tandy Center, Fort Worth, Texas 76102. Ask for Catalog TRS-80.

Radio Shack®
The biggest name in little computers

Prices May Vary at Individual Stores and Dealers

COMPLEAT COMPUTER CATALOGUE

We welcome entries from readers for the "Compleat Computer Catalogue" on any item related, even distantly, to computers. Please include the name of the item, a brief evaluative description, price, and complete source data. If it is an item you obtained over one year ago, please check with the source to make sure it is still available at the quoted price.

Send contributions to "The Compleat Computer Catalogue," *Creative Computing*, P.O. Box 789-M, Morristown, NJ 07960.

BOOKS AND BOOKLETS

SOFTWARE INDEX

A great quantity of software has been published in microcomputer books and magazines since late 1975. These programs range from complete BASICS to assemblers, routines, games and direct application programs. The SSI, an index to published microcomputer software, indexes hundreds of published microcomputer programs. The SSI is complete with cross references and over 130 program divisions. In addition, the SSI features complete publication documentation and bibliographic data. The January-July 1978 issue of the SSI, covering magazines from Jan 1976 to Sept 1977, plus many books, is $5.

Schreier Software Index, 4327 East Grove Street, Phoenix, AR 85040.

CIRCLE 180 ON READER SERVICE CARD

COMPUTER DATA DIRECTORY

Computer Shopping? *The Computer Data Directory* announces the first edition of its comprehensive catalogue for small-computer users. Now, from several hundred computer related firms, you can choose products and services from the largest list ever printed. Included are brand-name manufacturers of systems, peripherals, and accessories. In software, find who handles languages, business applications, household control, games, even custom programming services! Miscellaneous includes where to find books, magazines, newsletters, home-study courses, data banks, tools, and repair services. Computer stores and clubs are indexed geographically. $4.98.

The Computer Data Directory, P.O. Box 598, Cleveland, Ohio 44107.

CIRCLE 181 ON READER SERVICE CARD

MAGAZINES, JOURNALS

SMALL BUSINESS COMPUTER NEWS

Published monthly, *Small Business Computer News* is aimed mainly at present and future users of mini- and micro-computers and relevant peripherals. Each issue opens with news items, followed by a spotlight section, usually on a particular system that's discussed in detail, winding up with a casebook application study. $51 a year.

Small Business Computer News, Management Information Corp., 140 Barklay Center, Cherry Hill, NJ 08034. (609) 428-1020.

CIRCLE 182 ON READER SERVICE CARD

VENDOR LITERATURE

SYBEX CATALOG

This 16-page catalog describes the books and self-study courses available from Sybex. Books carry titles such as "An Introduction to Personal and Business Computing," "Microprocessors, From Chips to Systems," "Microprocessor Interfacing Techniques," and "Microprocessor Encyclopedia." The courses consist of a book and several cassettes, and cover areas such as "Introduction to Microprocessors," "Programming Microprocessors," and "Bit-Slice."

Sybex Inc., 2161 Shattuck Ave., Berkeley, CA 94704. (415) 848-8233.

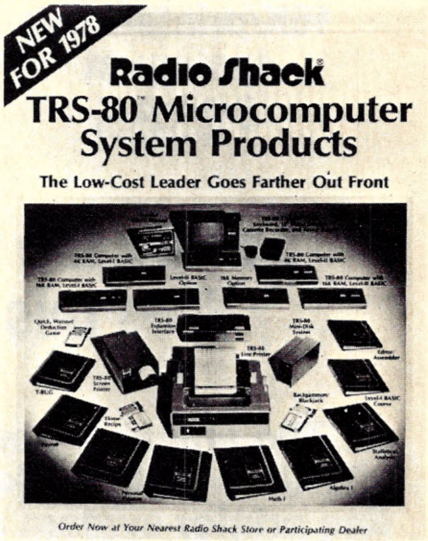
CIRCLE 183 ON READER SERVICE CARD

OHIO SCIENTIFIC CATALOG

The Spring 1978 catalog from Ohio Scientific gives details on a variety of BASIC in-ROM computer systems from the popular Challenger IIP up to the dual-disk Challenger III, plus several simple systems that permit getting started on a limited budget, the 74-megabyte hard-disk drive, as well as bare boards, populated boards, peripherals, and software.

From your OSI dealer, or Ohio Scientific Inc., 1333 S. Chillicothe, Aurora OH 44202.

CIRCLE 184 ON READER SERVICE CARD



RADIO SHACK MICROCOMPUTER CATALOG

Radio Shack's new 8-page TRS-80 Microcomputer System Products catalog features their $599 TRS-80 Microcomputer System and provides information on upgraded , systems, peripherals and ready-

to-use software developed specifically for the TRS-80. The basic TRS-80 system, described as the "beginner's choice," offers Level-I BASIC with 4K of ROM to produce a thorough and easy to understand computer language. Its 4K RAM is said to contain sufficient memory to accommodate many home, school, lab or small business uses. Expanded TRS-80 systems, including a 4K "Educator" system, priced at $1198, a 16K "Professional" system selling for $2385 and a 32K "Business" system for $3874 are also featured in the catalog. Also included in the new catalog is information on "How to Expand Your Existing TRS-80 System," with details of Level-II BASIC.

The new Radio Shack TRS-80 Microcomputer System Products catalog is available free, on request, from Radio Shack stores and dealers, nationwide.

ORGANIZATIONS

SC/MP USERS' GROUP

A users' group has been formed for hobbyists who have, or plan to acquire, a SC/MP or SC/MP-II computer system. No dues or fees are involved; the only cost to members is the self-addressed, stamped envelopes sent in by members for receiving the monthly newsletter. There will be a library of software and hardware information, to be made available to members on a cost basis. A bibliography is being compiled. If enough interest is shown, a homebrew system may be constructed, based on the SC/MP-II and using a common bus structure. Several newsletters have already been published.

Coordinator: Tom Bohon, 2215-A Walker Drive, Omaha, NE 68123.

DG USERS' GROUP

Attention Digital Group system owners! You are not alone. There are others out there who would like to share their joys and sorrows with you. An independent User's Group is being formed to act as a clearinghouse for exchange of information. The first issue of the newsletter features an evaluation of Micro-Com software, a Selectric interface (hardware and software), a discussion of problems in expanding past 26K, a flea-market section, and much more...

For information write to: DG Users' Group, P.O. Box 316, Woodmere, NY 11598.

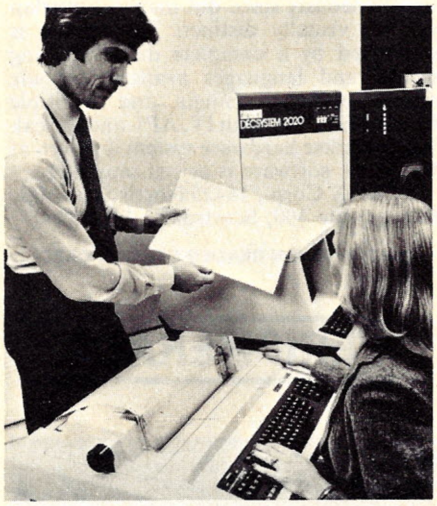
CIRCLE 185 ON READER SERVICE CARD

CENTRAL STANDARDS LIBRARY

To help solve some of the standards problems in the hobbyist computer and microcomputer field, ALF Products is sponsoring a Central Standards Library as a means of standards information exchange for manufacturers, consumers, hobbyists, and others interested in standards. The Library will collect submitted standards and distribute them on a non-profit basis. Manufacturers currently participating include: ALF Products, IMSAI Manufacturing, PolyMorphic Systems, Proko Electronics, Vector Graphic, and Video Terminal Technology.

For more information on available standards, on how to submit standards, and on the Library's services, send $1 (to cover printing and mailing costs) to: The Central Standards Library; c/o ALF Products Inc.; 128 S. Taft; Denver, CO 80228. You will receive a copy of the first CSL Newsletter and the first submitted standard (a parallel interface standard).

COMPUTERS



DECSYSTEM-2020—LOWEST-PRICED MAINFRAME

Digital Equipment Corporation has announced the world's lowest-priced mainframe, the DECSYSTEM-2020. The new computer has a minimum configuration price of $150,000 and features the lowest power consumption of any mainframe system. It is the first mainframe that can operate completely independent from the traditional "computer room" environment.

The DECSYSTEM-2020 is a general-purpose computer system and offers concurrent interactive timesharing, multi-stream batch, and transaction-oriented processing.

The configuration for the DECSYSTEM-2020 Central Processing System with the TOPS-20 operating system ranges from a memory of 128K 36-bit words (512K bytes) to 512K words (2 megabytes); one 67-megabyte magnetic disk or one 176 megabyte disk to eight of either; up to four 800/1600 bpi 9-track tapes; and 8 to 32 asynchronous lines. A full line of software products is optionally available, including ANSI-standard COBOL, FORTRAN IV, BASIC-Plus-2, ALGOL, APL, CODASYL-standard Data Base Management System (DBMS), an Interactive Query Language (IQL), and CPL (an interpretive subset of PL/1).

Digital Equipment Corp., Maynard, MA 01754.

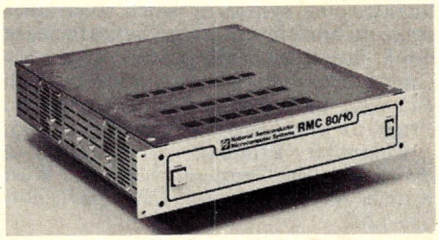
CIRCLE 186 ON READER SERVICE CARD

FOUR-DISK SOL SYSTEM

An integrated small computer system with four full-size floppy disks on-line has been introduced by Processor Technology Corp. The new system, Sol System IV, includes the company's Sol-20 mainframe with 50,176 8-bit words of RAM memory, a Helios II Model 4 Disk Memory System, PTDOS Disk Operating System, Extended Disk BASIC, a video monitor and complete documentation. Total mass storage capability on four formatted disks is 1.5 million bytes. The PTDOS Disk Operating System offers advanced functions including complex editors, assembler, device-independent files, and random indexed files. The video display can be addressed randomly to any position on the screen so one can easily write powerful forms-control procedures. Extended BASIC includes string and advanced file functions, timed input, complete matrix algebra, base 10 and rational logarithms, trigonometric functions, exponential numbers and 8-digit precision. Disk FORTRAN and Disk PILOT languages are low-cost options. The Sol System IV is $7995.

Processor Technology Corp., 7100 Johnson Industrial Drive, Pleasanton, CA 94566.

CIRCLE 187 ON READER SERVICE CARD



LOW-PROFILE RACK-MOUNTED MICROCOMPUTER

A low-profile 8-bit microcomputer, from the Computer Products Group of National Semiconductor Corporation, requires only three and one-half inches panel space in a standard 19-inch rack. Based on the BLC80/10 central processor board using the INS8080A microprocessor, the rack-mounted computer model RMC 80/10 microcomputer incorporates programmable serial and parallel input/output, complete busing, power supply, fans and three expansion-board slots.

The microcomputer has six general-purpose 8-bit registers, an accumulator, a 16-bit program counter and a 16-bit stack

pointer. The BLC 80/10 cpu has 1K bytes of on-board RAM, and four sockets for up to 4K bytes of PROM using MM2708 devices. The RMC 80/10 is $1345; the two-chip monitor is $200.

National Semiconductor Corp., Computer Products Group, 2900 Semiconductor Drive, Santa Clara, CA 95051.

CIRCLE 188 ON READER SERVICE CARD

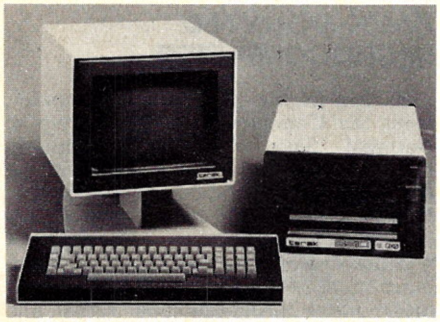
BUSINESS SYSTEM

A low-cost, microprocessor-based business computing system that combines accounting functions with word processing has been introduced by Computer Products of America, a division of The Computer Mart. The new system, ABACUS 1, is a complete hardware and software package designed to handle basic accounting for small businesses. ABACUS · 1 includes a Z-80 microprocessor, dual Northstar disk system, video display, keyboard and printer plus software.

Functions performed by the ABACUS 1 include general ledger accounting, accounts receivable, accounts payable, inventory, payroll, mailing lists, data entry, sorting and file management. A character-oriented word processing system is available as an option. The ABACUS 1 features an interactive, double-entry bookkeeping system in which receivables decrease book inventory, payables increase book inventory, and general ledger accounts are updated automatically with extensive and valid accounting controls. Prices for the ABACUS 1 start at $5,995.

Computer Products of America, a division of The Computer Mart, 633 West Katella Avenue, Orange, CA 92667. (714) 633-1222.

CIRCLE 189 ON READER SERVICE CARD



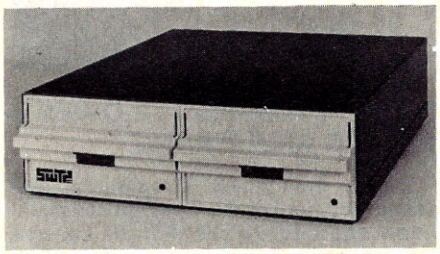
GRAPHICS COMPUTER

TERAK Corporation is now shipping a complete standalone table-top computer system featuring raster-scan graphics capability. Based on the company's 8510 data processor, the 8510/ a graphics computer system includes the Digital Equipment Corporation's LSI-11 microcomputer, a single floppy-disk drive and controller to handle up to four drives, 56K bytes of read/write MOS RAM, single asynchronous serial interface, video electronics, 12-inch CRT display and keyboard. Graphics are displayed in bit-map fashion using a 320-dot-wide by 240-dot-high matrix refreshed directly from any contiguous segment of main memory. Text is displayed as 24 lines of 80 characters each from a randomly addressable page buffer independent of main memory. Text and graphics are independently controlled, including three horizontal blanking zones in each mode of display. Text and graphics can be displayed simultaneously since dot matrices overlap yet are visually distinct. The 8510/a is supported by a complete disk operating system and languages available include macro-assembler, Single and Multiple User BASIC, Fortran IV, APL and Pascal. The complete hardware system is priced at $7,850.00; software is priced separately.

TERAK Corp., 14425 North Scottsdale Road, Suite 100, Scottsdale, AR 85260.

CIRCLE 190 ON READER SERVICE CARD

PERIPHERALS

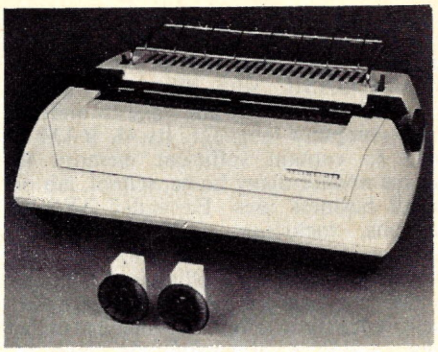


FLOPPY DISK SYSTEM FOR SOUTHWEST 6800

The Southwest Technical Products Corporation DMAF1 is a dual drive, single density, double sided 8-inch floppy disk system. The hardware consists of a SS-50 bus (SWTPC 6800) compatible DMA (direct memory access) controller capable of handling up to four drives, two CalComp 143M double-density disk drives, regulated power supply, drive-motor control board, cooling fan, diskette and interfacing cables. The supplied software includes "one of the most powerful and easy to use microcomputer disk operating systems yet available for either the hobbyist or businessman." An 8K BASIC Interpreter, with disk file capability and string functions, is also included with the system. Each diskette holds approximately 600,000 bytes (characters) of data, and with two drives you have over one megabyte of data on line. $2,095 assembled, $2,000 kit.

Southwest Technical Products Corp., 219 W. Rhapsody, San Antonio, TX 78216. (512) 344-0241.

CIRCLE 191 ON READER SERVICE CARD

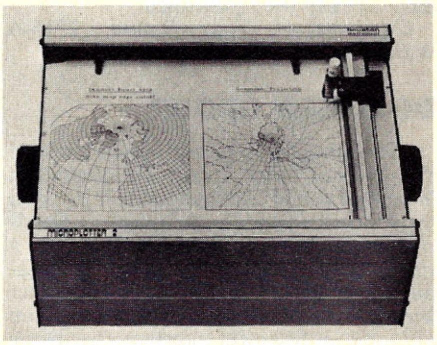


ALGORITHMICS PRINTER

The Algorithmics PR-DW1 Daisy Wheel Printer is a letter-quality printer unit designed for use with microcomputer systems for high-quality printing and plotting applications. This printer operates under control of an internal microprocessor and communicates with the host microprocessor over a high-speed asynchronous parallel interface. It prints bi-directionally at rates of 45 characters per second. The carriage can be positioned left and right in increments as fine as 1/120 of an inch and the platen can be rolled forward and backward in steps of 1/48 of an inch. Hardware options include 55 characters per second version, metal print wheel, cam-feed platen and forms tractor. Twenty-eight different type styles are available on easily changed plastic and metal wheels. Suggested retail price for the standard configuration including complete microcomputer interface and all software (text printing and graphics) is $2678.

Algorithmics Inc., Box 56, Newton Upper Falls, MA 02164. (617) 965-0545.

CIRCLE 192 ON READER SERVICE CARD



DIGITAL PLOTTER FOR PERSONAL COMPUTING

Imagine a digital plotter complete with RS-232C interface designed for the personal computing market that sells for only $1085. The Microplotter 2 is a true digital plotter with an 8½ x 11-inch page size, .005 or .01 inch resolution, and a RS-232C interface. Not a kit, it is complete and ready to plot. Houston Instrument has built the Microplotter 2 to provide the digital plotter which system builders have wished for, microcomputer users have desired, and personal computer lovers have envisioned. $1085.

Gabrielle C. Ryan, Houston Instrument, One Houston Square, Austin, TX 78753. (512) 837-2820.

CIRCLE 193 ON READER SERVICE CARD

OSBORNE & ASSOCIATES, INC.

The World Leaders In Microprocessor Books

If you want information on microprocessors, begin with the Osborne books.

ASSEMBLY LANGUAGE PROGRAMMING

8080A/8085 Assembly Language Programming

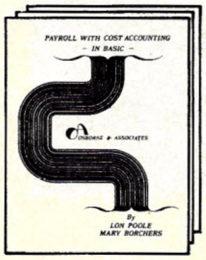
6800 Assembly Language Programming

These books describe how to program a microcomputer using assembly language. They discuss classical programming techniques, and contain simplified programming examples relevant to today's microcomputer applications.
#31003, 32003 (400 pages each)

PROGRAM BOOKS WRITTEN IN BASIC

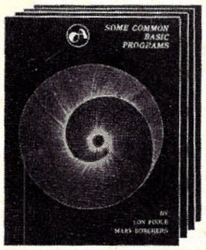
Payroll With Cost Accounting
Accounts Payable And Accounts Receivable
General Ledger

These books may be used independently, or implemented together as a complete accounting system. Each contains program listings, user's manual and thorough documentation. Written in an extended version of BASIC.
#22002 (400 pages), #23002*, #24002*

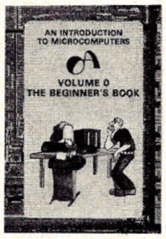


Some Common BASIC Programs

76 short practical programs, most of which can be used on any microcomputer with any version of BASIC. Complete with program descriptions, listings, remarks and examples.
#21002 (200 pages)

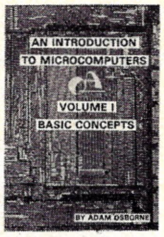


AN INTRODUCTION TO MICROCOMPUTERS



Volume 0 - The Beginner's Book

If you know nothing about computers, then this is the book for you. It introduces computer logic and terminology in language a beginner can understand. Computer software, hardware and component parts are described, and simple explanations are given for how they work. Text is supplemented with creative illustrations and numerous photographs. Volume 0 prepares the novice for Volume I. #6001 (300 pages)



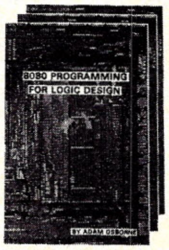
Volume I — Basic Concepts

This best selling text describes hardware and programming concepts common to all microprocessors. These concepts are explained clearly and thoroughly, beginning at an elementary level. Worldwide, Volume I has a greater yearly sales volume than any other computer text. #2001 (350 pages)

Volume II — Some Real Products (revised June 1977)

Every common microprocessor and all support devices are described. Only data sheets are copied from manufacturers. Major chip slice products are also discussed.
#3001A (1250 pages)

PROGRAMMING FOR LOGIC DESIGN



8080 Programming For Logic Design
6800 Programming For Logic Design
Z80 Programming For Logic Design

These books describe the meeting ground of programmers and logic designers; written for both, they provide detailed examples to illustrate effective usage of microprocessors in traditional digital applications.
#4001, #5001, #7001 (300 pages each)

OSBORNE & ASSOCIATES, INC. P.O. Box 2036 DEPT. L4 Berkeley, California 94702 (415) 548-2805 TWX 910-366-7277
9:00 a.m. - 5:00 p.m. Pacific Time

These prices effective July 1, 1978.	PRICE	QTY	AMT
6001 Volume 0 — The Beginner's Book	$ 7.95		
2001 Volume I — Basic Concepts	$ 8.50		
3001A Volume II — Some Real Products	$15.00		
4001 8080 Programming for Logic Design	$ 8.50		
5001 6800 Programming for Logic Design	$ 8.50		
7001 Z80 Programming for Logic Design	$ 8.50		
31003 8080A/8085 Assembly Language Programming	$ 8.50		
32003 6800 Assembly Language Programming	$ 8.50		
21002 Some Common BASIC Programs	$ 8.50		
22002 Payroll With Cost Accounting	$15.00		
	TOTAL		
	Sales Tax (Calif. residents only)		
	Shipping Charges		
	TOTAL AMOUNT ENCLOSED		

- $6\frac{1}{2}$%, SF Bay Area residents only
- 6%, California residents outside SF Bay Area
- **Payment by check or money order must be enclosed for orders of 10 books or less.**

NAME

ADDRESS

CITY

STATE ZIP PHONE

SHIPPING CHARGES: Shipping for large orders to be arranged.

- ☐ All foreign orders, $3.00 per book, for air shipment
- ☐ 4th class $0.35 per book (allow 3-4 weeks within USA, not applicable to discounted orders)
- ☐ $0.75 per book, UPS (allow 10 days) in the U.S.
- ☐ $1.50 per book, special rush shipment by air in the U.S.

Please send information on:

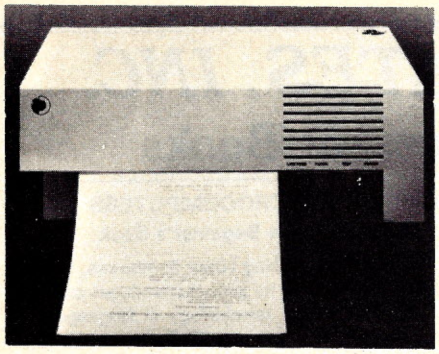
- ☐ Becoming an O&A dealer
- ☐ School discounts
- ☐ List of foreign distributors

***These books are scheduled to be published during 1978**
Please notify me when they are available:

- ☐ 23002 Accounts Payable and Accounts Receivable
- ☐ 24002 General Ledger

F8

CIRCLE 118 ON READER SERVICE CARD



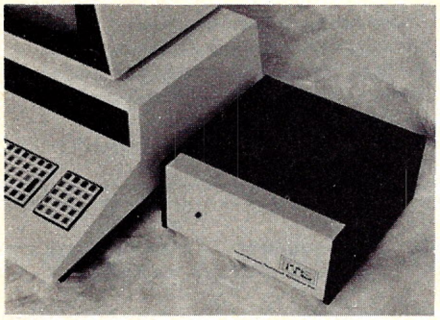
IMPACT PRINTER

A microprocessor-based RS232C - compatible impact printer has been introduced by Integral Data Systems. The Integral IP-125 features an RS232C serial interface, parallel TTL level interface and full upper and lower case ASCII character set (96 characters) as standard equipment. Capable of printing multiple copies on ordinary 8½-inch roll, fanfold or sheet paper, the microprocessor-controlled IP-125 incorporates a 256-character multi-line buffer to achieve an instantaneous print rate up to 100 characters per second with a sustained throughput of 50 cps at 80 columns per line. Format is 7x7 dot matrix, with a maximum line length of 132 columns.

Integral Data Systems, Inc., 5 Bridge St., Watertown, MA 02172. (617) 926-1011.

CIRCLE 194 ON READER SERVICE CARD

MISC. HARDWARE



EXTERNAL MEMORY FOR PET

The PEM-8K stand-alone 8K external memory system from International Technical Systems is compatible with either the 4K or 8K PET 2001 computer. Connection to the PET's existing memory is made through a three-foot interface cable and plug that mates with the PET's memory-expansion connector. A regulated power supply is included. As an introductory bonus, International Technical Systems offers a 10K software package that allows the PET to perform financial and investment calculations, hyperbolic trig functions, N! factorials and more. $279.

International Technical Systems, Inc., Box 264, Woodbridge, VA 22194.

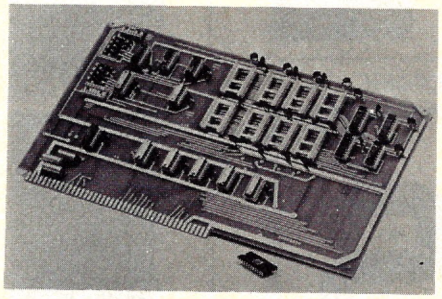
CIRCLE 195 ON READER SERVICE CARD

VRAM BOARD FOR SWTPC 6800

SWTPC 6800 owners looking for a memory-mapped video display device for their computers are advised that Gimix, Inc. has a VRAM board plug-compatible with the 6800. The Gimix board displays upper-case ASCII in a 16-by-32 (standard) or 16-by-64 format. 1024 bytes of memory for the display are provided on-board. Bit 8 (not used in 7 bit ASCII) is used to control cursors. A byte with bit 8 on will be represented as a cursor, while bytes with bit 8 off are represented by normal ASCII characters. A terminal replacement driver (which permits the VRAM to be used like a CRT) is provided with the board. The VRAM board can also be used for high-speed video graphics. $249.

Gimix, Inc., 1337 W. 37th Place, Chicago, IL 60609.

CIRCLE 196 ON READER SERVICE CARD

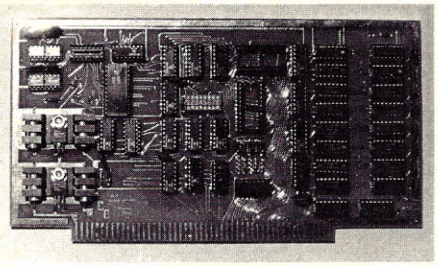


PROM BOARD

The Electronic Solutions PROM-8 is an 8K byte PROM board compatible with Intel's SBC 80/10 and National Semiconductor's BLC 80/10 single-board computers. The PROM-8 contains sockets for eight 2708 EPROMs. Address selection is done via jumpers. The PROM-8 board is divided into two 4K segments. Each segment's base address may be independently set by jumpers. Base addresses fall on 4K boundaries. $186.

Richard E. Van Antwerp, Electronic Solutions, Inc., 7969 Engineer Road, San Diego, CA 92111. (714) 292-0242.

CIRCLE 197 ON READER SERVICE CARD



CHARACTER GENERATOR BOARD FOR Z-80

Objective Design now offers a high-speed version of its super-dense graphics add-on board for S-100 systems — the Programmable Character Generator. This S-100 card can be used with the Processor Tech VDM or SOL, Polymorphic Systems VTI, Solid State Music Video Board, and other video boards using the Motorola family of 9x7 matrix generators. The user can create individual characters, store them in on-board RAM, and access the characters directly from the keyboard. Each character can be created on a maximum 8x16 matrix with a resultant screen density as great as 512x256. Special character sets can be created for music and speech notation, foreign alphabets, scientific applications, high-level computer languages, and games. Since characters can be stored in either the PCG RAM or regular system memory accessible from the bus, any number of characters can be displayed under software control, with a maximum of 128 different characters, either ASCII or programmed, on screen at any one time. By switching character sets in and out, hundreds of characters can be used in game programs or animation series. $165.95 kit, $215.95 assembled.

Objective Design, Inc., P.O. Box 20325, Tallahassee, FL 32304.

CIRCLE 198 ON READER SERVICE CARD

S-100 BUS TERMINATING BOARD

If you have a noisy S-100 bus or a board that intermittently has problems, then you may be interested in a board produced by Digital Micro Systems. It is a board that terminates each bus line with about a 190-ohm impedance. This matches the impedance of the output drivers on the bus and absorbs the ringing, reflection, overshoot, and noise that can sometimes be a real headache in a computer system. Large computers have used termination for years. $25.

Digital Micro Systems, Box 1212, Orem, UT 84057. (801) 224-2102.

CIRCLE 199 ON READER SERVICE CARD

LOW-COST KEYBOARD

The MKB-2 Keyboard by MicroAge, designed for use with the new 64- and 80-character display video-boards, combines the most popular keyboard features with a low affordable price. Included as standard in the MKB-2 are a numeric key pad, upper and lower case, cursor control keys, 2-key rollover, and auto repeat on all keys. Plus, the MKB-2 is assembled in a heavy duty steel case with parallel interface, strobe or pulse, on-board regulation (5v, 12v), complete with standard DB25S connector, and black double-injection molded keys. $149.

MicroAge, 1425 W. 12th Place #101, Tempe, AR 85281. (602) 967-1421.

CIRCLE 200 ON READER SERVICE CARD

EPROM PROGRAMMER

Smoke Signal Broadcasting has a new low-cost 2708 EPROM programmer. The POP-1 interfaces to the conpany's P-38-1 and P-38-FF EPROM boards, which are SS-50 bus-compatible products. Complete software is provided on audio cassette. An adaptive programming technique allows most 2708's to be programmed in 15 seconds instead of the usual one and a half minutes. A separate self-contained power supply is used for the programming voltage, insuring sufficient current capability to program EPROM's from any manufacturer. $149.

Smoke Signal Broadcasting, P.O. Box 2017, Hollywood, CA 90028. (213) 462-5652.

CIRCLE 201 ON READER SERVICE CARD

SCREENSPLITTER TV DISPLAY SYSTEM

Screensplitter is a complete TV text display hardware/software system for your personal or business microcomputer. The entire package comes on a single S-100 bus printed-circuit board, with onboard software. Plug it into your system, connect the coaxial cable to a 10-MHZ or better TV monitor, and you're in business. VIEWPORT displays 40 lines of 86 characters each on your screen, a total of 3440 characters in an onboard 4K display buffer. This buffer becomes part of your system's central address space, locatable to any 8K boundary. Writing a character on the screen is as simple as storing an ASCII code in Screensplitter's buffer. And you can write information as fast as you please, since Screensplitter doesn't make you wait. Each character may have its figure-ground reversed independently, since black-on-white, white-on-black is determined from the high order bit of each byte in the display buffer.

Also onboard is Screensplitter's 1K 8080 Window Package. This package of user-callable display functions also becomes part of your system's central address space, and includes all advanced functions for maintaining up to 3440 logically independent I/O "windows" on the VIEWPORT screen. You can have a single 86 by 40 window, 3440 1 by 1 windows, or anything in-between: from 1 to 3440 windows of various sizes, located at various positions on the screen.

Screensplitter is available in kit form, complete with all the CRT logic, 4K static display memory, and 1K onboard software (including other user-callable functions not described, at $329. Assembled, $429.

Micro Diversions, Inc., 7900 Westpark Dr., Suite 308, McLean, VA 22101. (703) 827-0888.

CIRCLE 202 ON READER SERVICE CARD

SOFTWARE

MAILING LABEL AND DIRECTORY LOOKUP

HSC Computer Services, has developed a Mailing Label and Directory Lookup Package (MLDLP), compatible with the North Star disk operating system. The package written in North Star BASIC allows:

(1) Entry of up to 480 records on each diskette. Each record consists of name, address and telephone number. Duplicate records are checked for.
(2) Listing of all records on file.
(3) Lookup of name in file with retrieval of address and telephone number.
(4) Printing of mailing labels. User may control what region to which labels are to be printed. Also available is a listing of North Star error messages and their meaning. The MLDLP package on diskette with documentation is $50. The error message listing is $5.

HSC Computer Services, Ltd., P.O. Box 43, Brooklyn, NY 11236.

CIRCLE 203 ON READER SERVICE CARD

MICROSOFT COBOL-80

Microsoft, the company that authored Altair BASIC and FORTRAN-80, now delivers COBOL-80, the first COBOL for 8080/Z-80/8085 microprocessor systems. COBOL-80 conforms to the 1974 ANSI standard, giving users immediate access to programs already written in COBOL. All Level 1 features and the most useful Level 2 options for the "Nucleus" and for Sequential, Relative and Indexed file handling facilities are included. Additionally, Level 1 Table Handling, Library and Inter-Program Communication facilities are provided. Of the advanced Level 2 features, Microsoft has included the verbs STRING, UNSTRING, COMPUTE, SEARCH and PERFORM (varying/until), along with convenient condition specification by way of condition-names, compound conditions and abbreviated conditions. COBOL-80 allows a packed decimal data representation to conserve memory on floppy disks.

The COBOL-80 system consists of two complete packages: a compiler for translating source code into relocatable object code, and a runtime system containing standard routines needed by the object code at execution time. The whole system may be run in less than 32K bytes. Rate of compilation is 250 lines per minute. COBOL-80 is available off the shelf to run under the CP/M and ISIS-II operating systems for just $750 per copy.

Microsoft, 300 San Mateo, NE, Suite 829, Albuquerque, NM 87108. (505) 262-1486.

CIRCLE 204 ON READER SERVICE CARD

FILE MANAGEMENT SYSTEM

KSAM is a file management system designed specifically for floppy-disk microcomputer systems. It was developed primarily for use in applications where large files are involved and fast random access is a necessity. Such applications include, but are not limited to, inventory control, reservation systems, library systems, accounts receivable and bill of materials processing. Random storage and retrieval of records is based on the contents of a user-defined data field within the record which is called the key. The key must be unique for each record and it can be any string up to 255 characters long. Examples of keys are: part numbers for inventory control, account numbers for billing systems and customer names for mailing list applications. KSAM80 also supports sequential access of records starting at any point within a file, random access by partial key and random access by relative record number. Sequential and random access commands can be intermixed freely. KSAM80 was originally developed under Zilog's Z80 OS 2.0 but can be easily implemented in many existing microcomputer operating systems. $195 for an end-user.

EMS, 3645 Grand Av., Suite 304, Oakland, CA 94610. (415) 834-4944.

CIRCLE 205 ON READER SERVICE CARD

CHESS IN BASIC

This chess game will challenge a beginner or expert, in standard BASIC; runs in 16K of free memory, can be used on Teletypes or CRTs. With complete documentation and a paper tape, $20.

Tenberg Software Systems, 10311-J Malcolm Circle, Cockeysville, MD 21030.

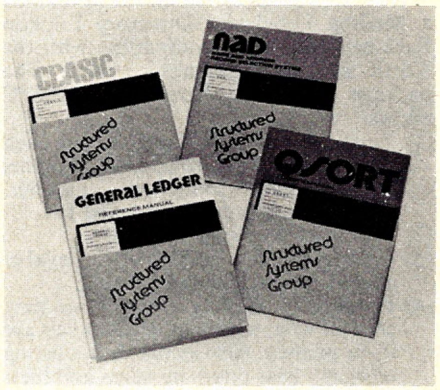
CIRCLE 206 ON READER SERVICE CARD

MULTI-USER/MULTI-TASKING OPERATING SYSTEM

Administrative Systems, Inc., (A.S.I.) has announced an upgraded version of their time-shared 8080 operating system, TEMPOS (Timed Environment Multi-Partitioned Operating System). The package is designed to be transportable among systems, through a new System Generation Routine. This Routine provides a mode during which the user interactively sets up TEMPOS for the configuration needed. Provisions are made (and extensive support documentation provided) to insert drivers for non-standard floppy disc drives (MITS and iCOM drivers are supplied), I/O devices (all MITS I/O boards and the IMSAI SIO2 are provided) and vectored interrupt-real-time clocks (again, MITS' VI/RTC and IMSAI's PIC-8 are both supported). The only real hardware requirements are an 8080 processor, 48K of RAM (addressed from location Ø), vectored interrupts and a real-time clock, with at least one "system console" I/O terminal. TEMPOS has many features of systems normally found only on much larger machines, for example, batch processing simultaneously with interactive use, foreground/-background processing, true multi-tasking (many different jobs/processes executing simultaneously) and programs like OPUS/TWO/(a high-level language with features of BASIC, FORTRAN and ALGOL), and TEXTED (A character-oriented text editor, which may be used to create OPUS/TWO or assembly language programs, or general-purpose ASCII text). TEMPOS is available on either MITS (hard-sectored) or iCOM (IBM 3740-type) diskette. $785. A User's Manual is $20.00.

Administrative Systems, Inc., 222 Milwaukee, Suite 102, Denver, CO 80206, (303) 321-2473.

CIRCLE 207 ON READER SERVICE CARD



BUSINESS SYSTEMS SOFTWARE

Structured Systems Group has announced its Business Systems Series. Designed to run on the 8080 or Z-80 CPU and the CP/M operating system, all software comes fully documented, field-tested and human-engineered for ease in operation. The comprehensive general ledger system is designed for professional accountants and small businesses. Quickly set up any custom chart of accounts to handle single or multiple departments. Interactively verify and customize data, formats, and headings. Written in CBASIC, $995. The Name and Address (NAD) System maintains files and allows selection on all fields for printing labels, reports, or new files. In CBASIC, $79. QSORT is a fast and efficient, easy to use, full disk sort/merge. Its automatic operation, multiple sort keys, and complete backup provide power and flexibility. In 8080 code, $95. CBASIC is an advanced, comprehensive, commercially oriented compiler/interpreter including full disk access, PRINT USING, 14 digits of precision, and more. With 85 page manual, $99.95.

Structured Systems Group, Inc., 5615 Kales Ave., Oakland, CA 94618. (415) 547-1567.

CIRCLE 208 ON READER SERVICE CARD

HIGH-SPEED RECORDING & LOADING PROGRAM

Shifting Sands Microcomputer Products' WHIZ is a software program for recording and loading MC6800-based programs on the SWTPC M68 computer with MIKBUG and the SWTPC AC-30 cassette interface. WHIZ operates at nine times the standard MIKBUG format 300-baud speed and three times that of the binary format. This higher speed capability is provided without any modifications to the standard computer or AC-30; just load and go. Load a 1K program in 14 seconds, 4K in 48 seconds, and 8K BASIC in 85 seconds. WHIZ includes an interactive front-end that allows the specification of a header and program-start address to be placed on tape for read-back later. WHIZ is supplied on Kansas City standard cassette in MIKBUG format and includes a built-in relocater to place your own copy of WHIZ in RAM memory wherever you wish. WHIZ is optionally available in 2708 EPROM. $15.95.

Shifting Sands Microcomputer Products Corp., Box 441, Fairborn, OH 45324.

CIRCLE 209 ON READER SERVICE CARD

COMPUCOLOR 8001 PROGRAMS

Raymond Schreiner has Compucolor 8001 programs for sale and swap with other interested owners. They include several programs for use with a light-pen and also contain a software clock for timed inputs. All are written in BASIC. Memory requirements vary from 1K to 10K and the programs are on Icom or Verbatim softsectored mini discs *only*. (floppy tapes are not compatable due to tape-head alignment differences).

For more information and a list of programs, send a *self-addressed stamped envelope* to Raymond Schreiner, 391 Broadway, Bayonne, NJ 07002. (201) 858-0799.

CIRCLE 210 ON READER SERVICE CARD

DATA ENTRY AND DEMAND REPORTING SYSTEM

Selector II is a data entry and demand reporting system for users of CP/M and Microsoft Extended Disk BASIC. It provides timely information at the user's fingertips, as well as allowing on-line updating of data files. Unique reports can be generated in minutes, with no programming experience required. The required information is simply selected with any variety of conditions desired and in the order wanted. Output can be used by programs, displayed or printed as a titled, columnized report. One time license price of $225 includes one-year warranty. Selector II is available on single-density 8-inch diskette complete with user's manual.

Micro-Ap, 8939 San Ramon Road, Dublin, CA 94566. (415) 828-6697.

CIRCLE 211 ON READER SERVICE CARD

MULTIUSER BASIC

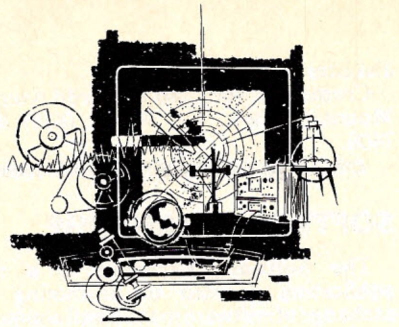
Midwest Scientific Instruments announces a multi-user BASIC for the MSI-6800 computer. The BASIC, based on the 6800 BASIC by Robert Uiterwyk, supports up to four users, and is completely compatible with MSI's single-user disk BASIC, including data files. The multi-user BASIC permits several users to access the same data file for input simultaneously, and in a 56K system gives four users the effect of using a single-user 24-28K system. No word on price yet since the multi-user BASIC is very new.

Midwest Scientific Instruments, 220 W. Cedar, Olathe, KS 66061.

CIRCLE 212 ON READER SERVICE CARD

GRAPHICS FOR SOUTHWEST

GRAPH#1 and GRAPH#2 are utility packages enabling convenient use of the SWTPC GT-61 graphics display with a 6800 computer. GRAPH#1 is for use with a machine-language program while GRAPH#2 is for use through SWTPC 8K BASIC version 2. When using GRAPH#2, the user can program the GT-61 display directly from a BASIC program without any machine language programming. Both versions can display (as well as erase) points, lines, and the complete upper-case ASCII character set. Additional capability for user-defined characters (such as chess symbols, or even space ships) adds to the ease of use and versatility of GRAPH. Applications for this package include display of statistical data, complete with legends. Game possibilities are numerous, since real-time moving displays can be



created, even from BASIC. $5.50 for paper tape; $6.50 for Kansas City Standard Cassette.

Applied Microcomputer Systems, Box 68, Silver Lane, NH 03875. (603) 367-8004.

NEW H8 SOFTWARE FROM HEATH

Heath Company has introduced additional software for its H8 Personal Computer. Extended Benton Harbor BASIC with file capability, the manufacturer says, is a faster, more powerful version of the BASIC software provided with the H8. It includes, among other things, character strings, more convenience and math functions, dynamic storage allocation and access to a real-time clock. Extended Benton Harbor BASIC requires 12K to 18K of memory and is available in audio cassette ($20) or paper tape ($10) form. New games software for the H8 includes PA-82 Biorhythm, PA-83 Space War and PA-84 Game Set #1 that incorporates Craps, Orbit, Tic Tac Toe, Nim, Hexapawn, Hangman, Hmrabi and Derby. Heath says Biorhythm runs under Extended Benton Harbor BASIC and requires 16K of RAM. Space War and Game Set #1 require 24K and 8K of RAM respectively. The games are available in cassette form only and sell for $10 each (mail-order, Benton Harbor).

Heath Company, Dept. 350-580, Benton Harbor, Michigan 49022.

CIRCLE 213 ON READER SERVICE CARD

WORD PROCESSOR FOR NORTH STAR DISK

The Electric Pencil character-oriented word processing software allows text to be entered and manipulated at any location as a continuous string of characters. Lines are formatted automatically. Features include cursor controls, global search and replace functions, bi-directional scrolling, right and left justification, wraparound text as well as character, line or block insert and delete. Page formatting is at the complete control of the user. The Electric Pencil has its own DOS. Storage and retrieval of text is simple and automatic. Six versions are available for North Star including SOL-20, VDM-1 or VTI videos, standard printer (TTY,etc.) or Diablo HyTerm. Minimum system hardware for 8080 or Z-80 micro is 12K of memory, printer plus interface, video monitor plus VDM-1 or VTI interface. North Star Disk and/or Tarbell or CUTS cassette interface and recorder. $100 for Teletype, or similar. $150 for Diablo Hy-Term.

Available from local dealers or from Michael Shrayer; 3901 Los Feliz Blvd., Los Angeles, CA 90027.

CIRCLE 214 ON READER SERVICE CARD

UNIVERSAL BASIC COMPILER

Futuredata has announced its Universal BASIC Compiler for use with 8080, Z-80, 8085, and 6800 microprocessors. The compiler includes standard BASIC statements, character strings, arrays, bit functions, as well as PEEK, POKE, INP, and OUT for operations at the machine level. Assembly-language statements may be intermixed with BASIC. A Debugger allows the user to set and clear breakpoints with BASIC statement numbers and to display BASIC variables. The compiler must be used in a Futuredata disk-based system with at least 32K of memory, and costs $300.

Futuredata Computer Corp., 11205 So. La Cienega Blvd., Los Angeles, CA 90045.

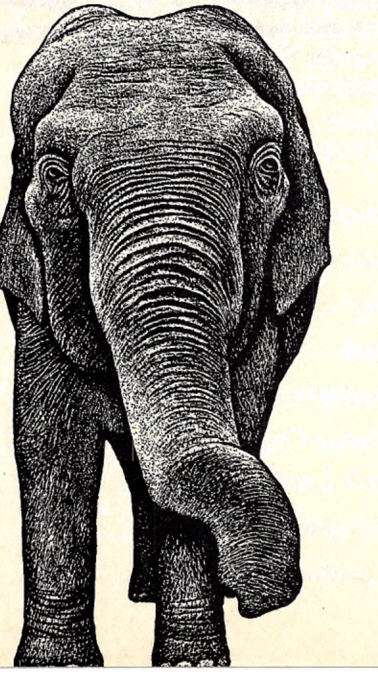
CIRCLE 215 ON READER SERVICE CARD

CROMEMCO SYSTEM SIMULATOR

Cromemco's TRACE is a powerful system simulator program designed to facilitate assembly-language program development on Cromemco computer systems. TRACE emulates the behavior of a Z-80 processor as it follows the logic of the user program. Virtually all aspects of System Operation can be simulated, including prioritized interrupts and I/O commands. TRACE options include control of register display and choice of display frequency. A historical record of program execution is maintained in a 100-instruction circular queue. Features that help the user locate errors quickly include warnings if the user writes to unexpected areas, simulation of input-output commands on the console, warnings of attempts to execute undefined commands, undefined calls to CDOS routines, improper return from subroutine calls, and execution of branch instructions or decimal adjust if the relevant flags are in an undefined state. The advanced features of TRACE enable it to be used in place of logic analyzers or in-circuit emulators in program development. TRACE is available on 5-inch (Model TSS-S) or 8-inch (Model TSS-L) IBM-format, floppy diskettes for $95.

Cromemco, Inc., 280 Bernardo Avenue, Mountain View, CA 94040. (415) 964-7400.

CIRCLE 216 ON READER SERVICE CARD



SOFTWARE EXCHANGE

The Software Exchange is a new publication devoted to promoting the exchange of software in the small computer marketplace. Reviews, articles and advertisements will focus on information useful to people who are interested in putting their computers to work. The Software Exchange is a bi-monthly magazine available soon at computer stores for $1.50 per issue and by subscription for $8 per year (six issues).

The Software Exchange provides reviews of user's groups, software packages, and consultants working with small computers. In addition, classified advertisements for computer software, for sale or wanted, are provided. Each program has a description of its operation, hardware requirements, and where the provided materials can be obtained.

People with software to sell or trade, and those looking for software to buy, can place classified advertisements in The Software Exchange for a nominal $2 fee.

The Software Exchange, Box 55056, Valencia, CA 91355.

MISCELLANEOUS

PET COMPUTER SERVICES

Two free PET services are available through the Microcomputer Resource Center: (1) The PET Cassette Exchange. Expand your program library easily. Exchange programs for the PET computer on cassette for free, no service charge. (2) The Ultimate PET — Resource Handbook. A continually updated listing of all hardware and software sources for the PET. Send a self-addressed stamped envelope for your free copy.

Inquiries: Len Lindsay, Director, Microcomputer Resource Center, Inc., 5150 Anton Dr. Room 212, Madison, WI 53719.

COMPUTER DAY CAMP

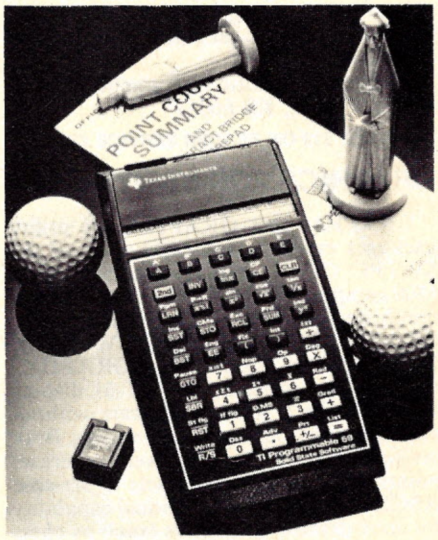
A summer computer camp is offered for junior and senior high students who have a keen interest in learning about computers. Students will learn to program in BASIC using mathematical and nonmathematical examples. Emphasis will be placed on a "hands-on" approach by providing each participant with ample computer time. There will be six Wang computers with screens, cassettes, printer and card reader available for the twenty participants to work individually and in small groups under the direction of an experienced staff.

The camp will be held daily from 9 AM to 4:30 PM at the Amity Regional Junior High School in Orange. There will be two separate two-week sessions running from July 5 to July 18, and from July 19 to August $, 1978. Participants may attend either or both sessions, which are $190 each.

For further information and an application, contact Dr. Michael Zabinski at (203) 795-9069 or write 382 Hitching Post Drive, Orange, CT 06477.

CIRCLE 217 ON READER SERVICE CARD

CALCULATORS



NEW LEISURE PROGRAMS FOR TI PROGRAMMABLES

To extend the problem-solving power of its programmable calculators for off-the-job enjoyment, Texas Instruments has introduced a new library of Solid State Software programs for use during leisure time. The new Leisure Program library for the TI Programmable 58 and 59 calculators contains 20 different programs useful to golfers, bowlers, chess players, football fans, bridge players, photographers and others interested in using the calculator for entertainment. In two of the programs, the alphabetic character and plotting capabilities of the new calculators and PC-100A print unit can be used to create computer-type art or store and write out messages. For golfers, one program in the Leisure Library cuts down the laborious task of figuring USGA handicaps. The football program forecasts a score based on two team's past performance; a second program allows two players to quarterback their teams and the print unit determines movement of the ball and keeps score. The bowling program keeps score for up to 90 bowlers who are bowling simultaneously with individual scoring possible in any order. The chess program computes ratings for chess players. Among other entertaining diversions in the library are programs to calculate biorhythms, land a spacecraft safely on Mars, simulate sea battles and play Blackjack, Acey Deucy, Craps, Nim, and others.

The plug-in program library module and manual have a suggested retail price of $35.00.

Texas Instruments Inc., Inquiry Answering Service, P.O. Box 53 (Attn: Leisure Library), Lubbock, TX 79408.

CIRCLE 218 ON READER SERVICE CARD

A Creative Computing Equipment Profile...

COMMODORE PET

Ludwig Braun

Commodore has been delivering their PET computer to customers since about September, 1977. For $595 you get a computer with 4K of RAM for user space plus 14K of ROM loaded with BASIC and an operating system. The machine is complete with a keyboard, a cassette recorder, and a nine-inch video monitor. If you pay $795, you get the same machine with 8K of user RAM. All of these are built into a single cabinet, so there are no cables to connect. This means that all you have to do to get the machine up is plug it in and turn it on. BASIC executes in ROM, so you don't use any of your RAM for BASIC, and BASIC is there ready to use immediately. You do lose 1,024 bytes of RAM for scratch pad, the stack, input buffers, etc., but the rest is available for user program, variable, and array space.

On balance, the PET is the best machine I've ever seen for the teacher and the student. (This doesn't mean it's perfect, as I will point out.) I feel this way because it is inexpensive, portable (I carry mine all over the U.S. to meetings and demonstrations), reliable, and easy to use. It also has an excellent BASIC.

BASIC

The BASIC is similar to MITS 8800 BASIC. It has strings (LEFT$, RIGHT$, MID$), graphics, and cursor-controlled editing. In addition, PET BASIC permits integer as well as floating-point representation of numbers and has ten-digit internal accuracy. It also has a better variable-naming convention than most BASICs. It is possible to define a variable named ACTIVITY, which the machine will accept and recognize. This means that BASIC program variables in PET can have more mnemonic significance than is possible in most BASICs. Actually, the interpreter recognizes only the first two characters of the name, even though it stores the entire name in the program. If, for example, the user enters the program

```
10 ACTIVITY=5
20 ACE=0
30 PRINT ACTIVITY, ACE
```

When the program is RUN, it will print

```
0          0
```

that is, both variables are treated as the variable AC, which is set to 0 by line 20.

Ludwig Braun, Dept. of Technology and Society, College of Engineering and Applied Sciences, State Univ. of New York, Stony Brook, NY 11794

If the user asks for a LIST, however, the program will be listed exactly as it appears above. This is useful in commenting within a program and should eliminate the REM statements we have had to use in documenting programs in the past.

Although the program line may be 80 characters long, the screen displays only 40 characters on a line (with 25 lines displayed). Longer lines overflow to the next line. This is a serious deficiency in the PET. For plotting graphs, or even for text, 64 or 80 characters is far better.

Graphics

The graphics capability is impressive in a machine at this price. There are 64 graphic characters in addition to the normal letters, numbers, and punctuation. Each printable character may be displayed direct (white on black) or reverse (black on white), essentially giving us 128 graphic symbols. These graphic symbols range from thin lines through rectangles, segments of circles, intersecting lines, and bars to the spade, diamond, heart, and club of playing cards. These symbols may be seen in the upper-case positions on the keys in the figure.

The PET may be put into a special mode in which the graphic symbols over the alphabetic characters are replaced by the lower-case alphabetic characters.

Cursor Control

The cursor control is an excellent feature for two reasons. The cursor may be moved up, down, right, or left by hitting the cursor controls. This permits powerful program editing. After a line has been entered, you can change an E to an R merely by moving the cursor to the E, hitting the R, and then hitting Return. The modification is made on the screen and in the stored version of the program. Using the cursor controls plus an Insert and Delete key, you can insert or delete words anywhere in a line after it has been entered; for example, if you inadvertently enter

```
50 IF X=10    260
```

(you forgot the THEN), you can place the cursor on the 2 of 260, insert four spaces, and then enter THEN. After Return, the line is changed. This editing capability means that long lines don't have to be re-typed. The second reason for excitement about cursor control is that the cursor controls may be imbedded in a program. This essentially gives us a two-dimensional TAB function, which together with the graphic symbols permits the display of complex visuals. And because you can blank the screen under program control, you can generate simple computer animations. Not only can we make programs more interesting for the user with graphics, but we can challenge kids to create their own graphics without requiring any knowledge of mathematics (see, for example, the DRAW program published in the November-December 1977 issue of *People's Computers*).

Cassette Recorder

The built-in cassette recorder permits the user to save programs merely by typing SAVE "NAME" where NAME is a real name (not A, B, or Q as in some micros). Re-entering requires only typing LOAD "NAME." The PET also has a VERIFY command which checks to see that the cassette version is identical to the one in memory. If it isn't, the screen displays a verify error. This feature is very valuable and is unavailable on most micros.

The user may customize BASIC because PET BASIC has a USR function to permit entry of user-defined machine-language routines.

Interfacing

The PET can be interfaced to the real world through any one of four external ports which are accessible through BASIC commands. There is an IEEE 488 interface, an eight-bit parallel port, a port for a second cassette recorder, and a port that brings out the system bus (it's not an S-100 bus). The second cassette recorder permits the user to put together a file management system, although the manual makes the user nervous about making a mistake and wiping out entire files.

Keyboard

The keyboard is, perhaps, the machine's most vulnerable component. It looks more like it belongs on a calculator than on a computer. The key-to-key spacing is only two thirds that of a conventional keyboard, which means that touch typing is difficult and even a two-finger duffer like me hits the wrong key now and then. On the plus side, the keyboard includes a number pad and all the common symbols (+, -, *, parentheses, quote mark, etc.) are available without requiring a shift.

Commodore's PET is one of the very few personal computers that combines all four basic units (keyboard, computer, cassette drive, video output) in a single package.

6502 Microprocessor

Commodore uses the MOS Technology 6502 (made by one of their subsidiaries), which has a very awkward instruction set. This means that the USR(X) function is more difficult to use than is true for most eight-bit microprocessors. This difficulty is obvious only with the USR function. In program execution, the PET is at least as fast as any other microcomputers which I have tried.

Deficiencies

The PET is a great machine, but it has some deficiencies in addition to the short screen line (40 characters) and the small keyboard. Some of these could have been overcome at almost no additional production cost but were overlooked by Commodore. Among these deficiencies are:

1. There is no composite-video signal available to the user. This means that a teacher who wants to use the computer in his/her classroom cannot display the screen on a monitor. Marc Hertzberg in our lab at Stony Brook has designed a system modification which provides this signal and which costs about $3 worth of parts and may be installed in less than half an hour.

2. There is no handle on the PET. I put on a $1 hardware-store handle and take my PET everywhere. Actually, none of the microcomputer manufacturers has put on a handle. Incidentally, the PET weighs 44 pounds, which is a bit heavy. Twenty pounds is a good weight goal.

3. We have a small number (three or four) of system crashes a month on each of our three PETs. We haven't been able to pinpoint the cause but suspect that line spikes are getting through the power supply. Fortunately, because BASIC is in ROM, we can recover by turning the machine off and then on again. Of course, in the process we lose whatever was in RAM. It would be nice to have a system restart mechanism which doesn't zero the memory.

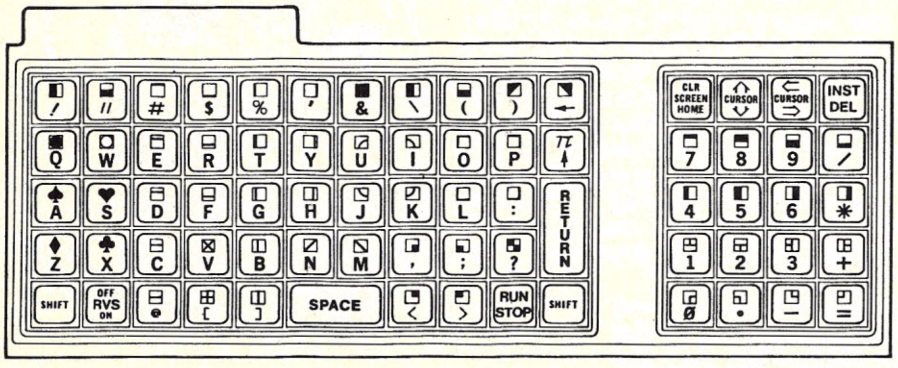
4. The cassette recorder doesn't have a counter, which is a real nuisance. If you have several programs on one tape, it is impossible to fast-forward to a point near the beginning of the program before loading. This means that you must start at the beginning of the tape and let it run at regular speed until it finds and loads your program.

5. The advertised transfer rate on the cassette recorder is 1000 baud. The effective transfer rate is closer to 250 baud because of overhead in the file format (leader space, redundancy, etc.). This, combined with (4) above, means that you must put only a small number of programs on a cassette or wait perhaps several minutes to get a program loaded. Arthur Leuhrmann of the Lawrence Hall of Science has solved this problem by using C5 cassettes and recording only one program per cassette.

6. Probably the most inexcusable deficiency in the PET is the sad state of the user's manual. When we got our first PET in October, we got a small booklet which told us how to turn the PET on, how to save and load programs, and which merely *listed* all the BASIC commands. Our second and third PETs arrived in December with a slightly larger booklet. This one gave brief illustrations of most of the commands and a little trouble-shooting information but still was inadequate. In January, we got about 50 loose-leaf pages describing the BASIC commands in adequate detail. With patience, I suppose that we eventually will get a real manual.

7. Another possible problem with the PET, at least for school administrators, is a lack of approval by Underwriters Laboratories. This doesn't mean that the machine is unsafe, but it is a base which Commodore should touch as soon as possible.

8. Additional memory is expensive. At a time when good 16K RAM boards sell for less than $400, charging $200 for 4K of RAM is hard to take. Additional RAM, beyond 8K, must be placed outside the main cabinet, which



Since each of the 64 graphics characters can be displayed in white on black or in black on white, 128 graphic symbols are available, including the four playing-card symbols.

The Rise and Fall of Bar Code

Steve North

Back in the May-June 1977 issue of *Creative*, we announced that we were going to start carrying programs in optical bar code. None have since appeared in *Creative*, nor is it likely that any ever will. Here's why.

For some time, there has existed a critical need for a means to publish machine-readable software. This is painfully obvious if you, like most of our readers, have typed in one of the lengthier programs we publish. The process of keying in a long program is error-prone, boring, and most people would just rather do something else. So, nearly two years ago, *BYTE* magazine proposed optical bar code for publishing machine-readable programs. The hobbyist bar-code standard consists of strips of dark bars that represent binary ones and zeros. These strips of bar code contain data bytes and other information needed to load the bar code and verify the data bytes. The bar code is read with a scanning wand connected to a microcomputer, with a suitable loader program. The entire bar-code reading process is very software-dependent; that is, the bar-code hardware merely tells the computer whether it is over a light or dark spot. The loader program converts this information into data and loads it into the system's memory.

Well, anyone can see that the entire micro industry has *not* switched over to bar code. Since the development of bar code, only a 6800 assembler and some other smaller 6800 object programs have been published. We at *Creative* had planned to publish some BASIC bar-code programs, but only when a bar-code reader was commercially available. (This originally started out as a review of that bar-code reader, made by MicroScan of Natick, MA, but we felt we could hardly evaluate a particular bar-code reader when the entire idea of publishing programs in bar code is questionable.) There are quite a few reasons why bar code just hasn't made it. To name a few:

- Bar code is bulky. It takes up more space than a printed listing of a program, and the printed listing is still necessary for documentation purposes and for everyone who doesn't have a bar-code reader. With only a limited number of pages in a magazine, it's hard to justify using so much space on bar code.
- Bar-code reading isn't that fast — only 30 bytes per second, typically. The proponents of bar code envisioned that one would load the bar code once, and thereafter use the normal system mass-storage device (cassette or floppy disk) for loading the program.
- To date, the only bar-code programs published have been 6800 object programs of interest to only a minority of microcomputer users. This has made it quite difficult for bar code to catch on.
- To read BASIC programs in bar code, one would have to make modifications within BASIC to patch in the bar-code loader. This is just beyond the skills of many micro users, and is totally impossible when BASIC is run in permanent Read-Only Memory. And this does not even address the problems of deciding what BASIC to use.
- Until very recently, the only bar-code reader you could have was one you built yourself or a surplus job. The Microscan reader works nicely and requires only one bit on a parallel port, but the $99 price tag is a little steep

So, for these reasons and more, bar code will never make it. Radio Shack, Commodore, et al, will never offer a bar-code peripheral. At present, the de facto standards for software distribution are audio cassettes and IBM 3740-compatible diskette. There are too many audio-cassette standards in use for any to be a real standard, but Radio Shack, Commodore, APPLE, Tarbell, and Kansas City are the biggies. For people who can read the popular cassette standards, *Creative* and others will offer software on cassettes. For the rest of the world, we'll always have printed listings of programs that can be used by anyone, with or without a bar-code reader. ■

Translation of bar code: If u cn rd ths u cn gt a gd jb.

PET continued...

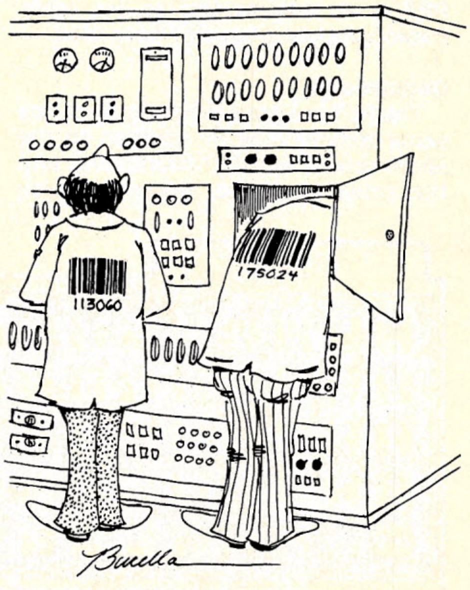
is a bit of a nuisance. This extra RAM isn't available yet, and no costs have been quoted.

9. The fact that the PET bus is not S-100 means that all the great boards (speech, music, Dazzler, A/D, etc.) cannot be used on the PET. All is not lost, however. At least two companies already offer S-100 adapters for the PET.

Conclusion

I don't want to quit on a downbeat. The bottom-line question is: Would I buy another PET? The answer is an enthusiastic "Yes!" It is the best classroom computer around on a price/performance basis and probably will be the standard of comparison for other personal computers we can expect to be introduced during the next couple of years. (Are you listening, TI?)

[Ed. note: As of the end of April, the 4K PET is not in production, although it may possibly resume in July. Another model of the PET, featuring a standard typewriter keyboard, is scheduled to go into production at the end of this year; it will not include a cassette drive, and some of the graphics functions will be dropped. A separate typewriter keyboard is also planned for later this year, for connecting to the present PET through its IEEE 488 interface.] ■



"The computer is rapidly taking the place of the ink blot. Both reveal more about the individual who reacts to them than they do about themselves."

John R. Coleman

ELECTRONIC SYSTEMS

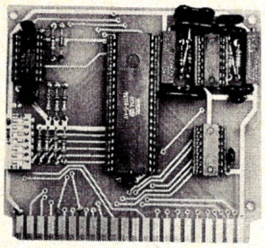
P.O. Box 9641 San Jose, CA 95157 (408) 374-5984

FOR CATALOG INCLUDING PARTS LISTS AND SCHEMATICS, SEND A SELF ADDRESSED ENVELOPE WITH 24¢ POSTAGE.

RS-232/TTY* INTERFACE NEW

Part no. 600

- Converts RS-232 to 20mA current loop, and 20mA current loop to RS-232
- Two separate circuits
- Requires +12 and -12 volts
- Board only $4.50, with parts $7.00

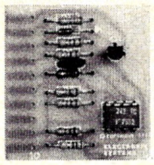


UART & BAUD RATE GENERATOR*

Part no. 101

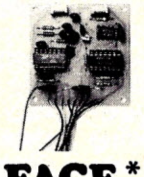
- Converts serial to parallel and parallel to serial
- Low cost on board baud rate generator
- Baud rates: 110, 150, 300, 600, 1200, and 2400
- Low power drain +5 volts and -12 volts required
- TTL compatible
- All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity.
- All connections go to a 44 pin gold plated edge connector
- Board only $12.00; with parts $35.00

RS-232/TTL INTERFACE*



Part no. 232

- Converts TTL to RS-232, and converts RS-232 to TTL
- Two separate circuits
- Requires -12 and +12 volts
- All connections go to a 10 pin gold plated edge connector
- Board only $4.50; with parts $7.00

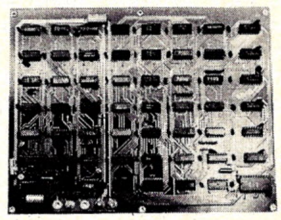


Part no. 111

TAPE INTERFACE*

- Play and record Kansas City Standard tapes
- Converts a low cost tape recorder to a digital recorder
- Works up to 1200 baud
- Digital in and out are TTL-serial
- Output of board connects to mic. in of recorder
- Earphone of recorder connects to input on board
- Requires +5 volts, low power drain
- Board $7.60; with parts $27.50
- No coils

TELEVISION TYPEWRITER



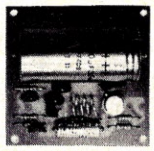
Part no. 106

- Stand alone TVT
- 32 char/line, 16 lines, modifications for 64 char/line included
- Parallel ASCII (TTL) input
- Video output
- 1K on board memory
- Output for computer controlled curser
- Auto scroll
- Non-destructive curser
- Curser inputs: up, down, left, right, home, EOL, EOS
- Scroll up, down
- Requires +5 volts at 1.5 amps, and -12 volts at 30 mA
- All 7400, TTL chips
- Char. gen. 2513
- Upper case only
- Board only $39.00; with parts $145.00

DC POWER SUPPLY*

Part no. 6085

- Board supplies a regulated +5 volts at 3 amps., +12, -12, and -5 volts at 1 amp.
- Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps.
- Board only $12.50; with parts $42.50 excluding transformers

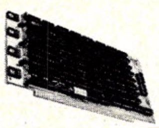


Part no. 107

RF MODULATOR*

- Converts video to AM modulated RF, Channels 2 or 3
- Power required is 12 volts AC C.T., or +5 volts DC
- Board $7.60; with parts $13.50

8K STATIC RAM



Part no. 300

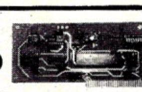
- 8K Altair bus memory
- Uses 2102 Static memory chips
- Memory protect
- Gold contacts
- Wait states
- On board regulator
- S-100 bus compatible
- Vector input option
- TRI state buffered
- Board only $22.50; with parts $160.00

TIDMA*

Part no. 112

- Tape Interface Direct Memory Access
- Record and play programs without bootstrap loader (no prom) has FSK encoder/decoder for direct connections to low cost recorder at 1200 baud rate, and direct connections for inputs and outputs to a digital recorder at any baud rate.
- S-100 bus compatible
- Board only $35.00; with parts $110.00

Apple II Serial I/O Interface*



Part No. 2

- Baud rates up to 30,000
- Plugs into Apple Peripheral connector
- Low-current drain
- RS-232 Input and Output

SOFTWARE

- Input and Output routine from monitor or BASIC to teletype or other serial printer.
- Program for using an Apple II for a video or an intelligent terminal. Board only – $15.00; with parts – $42.00; assembled and tested – $62.00.

MODEM*

Part no. 109

- Type 103
- Full or half duplex
- Works up to 300 baud
- Originate or Answer
- No coils, only low cost components
- TTL input and output-serial
- Connect 8 ohm speaker and crystal mic. directly to board
- Uses XR FSK demodulator
- Requires +5 volts
- Board $7.60; with parts $27.50

To Order:

Mention part number and description. For parts kits add "A" to part number. Shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. Dealer inquiries invited. 24 Hour Order Line: (408) 374-5984.* Designed by John Bell.

Apple II Computer

Steve North

With the state-of-the-art advancing so rapidly, many prospective micro buyers are looking for much more than a box with a CPU and some memory. All-in-one machines like the TRS-80 and PET, oriented towards computer *users,* not builders, are far outselling the older-style machines. In this exploding market for completely assembled consumer computers, the Apple II has some impressive features unmatched by many other micros.

The Apple II is based on the 6502 microprocessor, which has a number of devoted users (the 6502 is used in the PET, too). The Apple features a built-in ASCII keyboard, cassette interface, and a video interface, which is its most outstanding feature. The memory-mapped video interface displays both color graphics and normal alphanumeric data. The video interface may be connected to a color monitor, or to a regular color TV with an RF modulator. (M&R Enterprises makes such a modulator, which fits right inside the Apple and which requires absolutely no soldering. This is what we're using at *Creative.*) All the Apple's circuitry is contained on a single printed-circuit board that fits along the bottom of the Apple cabinet. Sockets are provided for extra memory (up to 48K of RAM) and extra I/O cards. 16K of memory is a reasonable amount to start with. The Apple also comes with two "game paddles" which are either one-dimensional joysticks, or knobs. Each game paddle also has a pushbutton switch. (Two more game paddles may be accommodated by the Apple but they are not provided with the standard unit.) To provide sound effects for games, primitive music synthesis, or just an audible signal, the Apple has a small built-in speaker under computer control. These peripherals are quite handy for many video games.

Built-In Software

The Apple II's built-in software (in ROM) includes a system monitor program, which is entered whenever the RESET key on the keyboard is pressed. This monitor lets you interact with the machine at a low level: read or save memory images on cassette tape, examine and change memory and CPU registers, go to a user program, etc. The monitor program also includes a "mini-assembler" which permits entry of machine-language programs by specifying a mnemonic and then the operands in hex. Better than nothing, I suppose, but not nearly as handy as a real assembler. The system monitor also has a built-in 16-bit processor simulator, called SWEET-16. This program, written in 6502 machine language, executes programs written in the language of a simple hypothetical 16-bit processor. Programs simulated this way are obviously a lot slower than they would be running on a real machine, but for some programming SWEET-16 can be used to write shorter and faster code.

The compact Apple II computer comes with a pair of game paddles and built-in integer BASIC. Among the many software options are Applesoft floating-point BASIC and the preprogrammed Checkbook.

Integer BASIC

Most Apple users will be more interested in the Apple's built-in integer BASIC. To enter BASIC from the monitor, just type control-B. Integer BASIC is quite complete and includes functions to access the color graphics, game paddles, etc. It doesn't have floating-point math, but it turns out that you can do a lot of things without floating point, especially in the area of graphics, where speed of execution is often a critical factor (hence the speedier integer math is in some ways desirable!).

Apple's integer BASIC does allow character strings, multiple statements, PEEK, POKE, CALL (for interfacing to assembly-language code), a TRACE mode, and other advanced features. Variable names are not limited to one or two characters, so they can be assigned to suggest the data they represent (for instance, RADIUS means a lot more than R1 when reading a BASIC program). Only the first two letters of a variable name are considered significant by BASIC, though. The game paddles are accessed by the PDL function. PDL(O), for instance, returns a value from 0 to 255 representing the position of game paddle #0. The GR statement tells the Apple to enable the graphics mode. In this mode, most of the display is reserved for color graphics, but the bottom three lines are used to display alphanumeric output (such as the execution of BASIC PRINT statements, INPUT, etc.). Furthermore, these two areas of the screen operate completely independently of each other — the normal scrolling of the bottom three lines of text does not affect the graphics area. I prefer this to the TRS-80's graphics, which force you to mix graphics and alphanumeric output on the same screen, so if the text has to scroll, so do the graphics. Before actually using the graphics, a COLOR statement is executed. This tells the Apple what color you want to plot in next. COLOR=9, for instance, means that you want to plot in orange. The colors are numbered from 0 to 15. If you don't want to use numbers, just say ORANGE=9 at the start of your program, and thereafter you can say COLOR=ORANGE. The COLOR statement doesn't restrict the display to one color; it's just that you plot graphics in only one color at a time. Once the color is set, you can draw some pictures! The screen is broken up into a 40x48 grid. PLOT X,Y will plot a particular point on the screen, while HLIN Y1,Y2 AT X1 and VLIN X1,X2 AT Y1 are used to draw horizontal and vertical lines. The SCRN function is used to examine the color plotted at any square on the screen. 40x48 graphics sound somewhat crude, but they're actually pretty good and are suitable for many nifty games, like Breakout (which comes with the Apple). And all this is available at the flick of a switch. (BASIC programs, of course, must be loaded from cassette).

Graphics Software

If you want more programming power, Apple has some other software products of interest. One of these is HIRES, a high-resolution graphics package, that permits plotting on a 160x280 grid, but only four colors (black, green, violet, and white) are allowed. You tell HIRES what to draw by POKE-ing parameters into fixed locations in memory. The HIRES

Users of Apple II can choose among a variety of preprogrammed software including Basic Finance, Checkbook, and High-Resolution Graphics.

routines can be used to plot a single point, a line, or a shape. To plot a shape, you have to set up a table of vectors which, given a starting location, tell the computer how to draw the shape. These vectors must be encoded in a binary notation that requires a little practice. HIRES is not only able to draw the shape, but also accepts a scaling factor (1 is normal size, 2 means twice normal, up to 255) and a rotation factor (0 means normal, 16 means rotate the shape 90 degrees, and so on). The HIRES package can be read from cassette tape, or put in a ROM if you use it frequently. 8K of memory must be reserved for the graphics alone.

Floating-Point BASIC

Applesoft BASIC is a full-feature floating-point BASIC, also loaded from cassette. You need at least 16K of memory to use it. Applesoft is a version of the popular Microsoft BASIC, with some special features for the Apple's graphics, generally very similar to those in integer BASIC. Applesoft does not have PDL functions, so to access to paddles, you need to use the PEEK function to read the paddle input port. Most Apple owners will probably want to use the Applesoft BASIC for serious application programs and more complex BASIC games, and integer BASIC for video games and simpler BASIC programming.

Documentation

The Apple documentation is variable, but more often than not, excellent. The integer BASIC manual, by Jeff Raskin, is outstanding. An absolute novice could pick up this book and begin using his Apple immediately, yet an experienced computer user won't find it frustrating. The Applesoft manual is rather skimpy, but it's intended as a short reference, not a text on BASIC. Nevertheless, there are a few BASIC example programs. A third manual, on the Apple itself (doubtless the first manual Apple had) attempts to cover everything — setting up the Apple, demo programs, integer BASIC, the system monitor, and the hardware including schematic diagrams. Actually, it isn't *that* bad, but it falls short in some important areas. For instance, the documentation on SWEET-16 consists of nothing but an assembly listing. I wouldn't even have known what the thing *is*, except that Steve Wozniak of Apple wrote an article for *Byte* on SWEET-16. At least they could get reprints. The documentation on HIRES was also a bit confused (the really intricate details on encoding the shape vectors were written by hand, yet). Obviously, Apple is in a transition phase with its documentation, and they've started with the documentation most important to new users. People who want to get into the system software are better equipped to piece out what's going on for themselves, and hopefully Apple will shortly bring the rest of the documentation up to the high standards of the BASIC manual.

The Apple II computer is a one-of-a-kind machine, probably the best micro on the market for color graphics. I also like the Compucolor machine, which has really superb color graphics and a light pen, but it's a little beyond my budget, and probably of many other consumer computer buyers, whereas the Apple is quite reasonably priced. There were a couple of things I didn't like about the Apple, too. The cassette interface did not like my cassette recorder, but I hear it works fine with Panasonic recorders. I would guess that the cassette works at 1200 baud, which is reasonably fast. The display is upper-case only, a throwback to the olden-daze of computers, but perhaps Apple had a reason for this. The display is rather narrow (40 characters) but you can compose lines longer than this by just typing past the edge of the line; the computer knows that it has to continue on the next line. Also, in integer BASIC, there is no way to slow down the listing of a program — it just keeps flying by. You can either list within a given range of line numbers just enough to fill the screen, or else hit RESET in the middle of a list, in which case you're back in the system monitor and have to type two more keystrokes to get back into BASIC with the program intact. One other complaint: the keyboard has cursor-right and cursor-left keys, which are very helpful in text editing with the screen, but no cursor-up, cursor-down, or home keys. Instead an "escape sequence" (the escape key and then another key) must be used to perform these functions. Also, the operation of the Repeat key is rather erratic; it should only operate when another key is depressed but sometimes pressing Repeat by itself causes the previous character typed to be echoed again. These are not really major problems, though.

Options

Options for the Apple II, besides more memory (all you need to expand the memory are a handful of ICs, which were called "Appleseeds" in one advertisement I saw) include interfaces for a printer, modem, and floppy disk. All of these are quite handy. The printer can be used to provide hard copy of BASIC programs or other stuff. The communications interface, used to connect a modem, would indicate that Apple sees networking of microcomputers as an important thing in the future, but to accomplish this a lot more will be required than hardware. It's a start, anyway. The floppy-disk option should also be interesting, since anyone who has used floppies can tell you they have it all over cassettes. It should be interesting to see whether floppy-disk attachments for computers like the Apple and TRS-80 enhance or detract from the integrated structure of hardware and software of these systems.

In short, the Apple is certainly one of the most versatile micros on the market. The user can view it as either a very sophisticated video game, a BASIC-speaking graphics machine, or a real nuts-and-bolts *computer*-computer. The Apple is not a machine for the classroom, or for the hobbyist who wants to use all the nifty S-100 boards, but for those especially interested in *using* a computer, even if they're just beginners, the Apple II is an excellent choice. ■

A Program for the Apple II...

high-resolution graphics for the apple II

Gary D. Dawkins

Description: This program allows the user to draw a shape in high-resolution graphics mode from the keyboard. The program simultaneously assembles a vector table for the shape in memory starting at a user-specified location.
Hardware requirements: Apple-II computer with at least 16K of RAM.
Loading the Program: This program calls the Apple high-resolution subroutines which, therefore, must be in the computer. Enter BASIC with control-B and set HIMEM to 8192. Load the program from the tape recorder in the usual manner. Start the program with RUN.
Using the Program: The program will clear the screen and ask you at what location in memory you wish to start assembling the vector table. Your answer MUST be a four-digit hexadecimal number. Use leading zeroes if necessary (for example, hex 950 is entered as 0950). If you made an error at this point, terminate the program with control-C and start it again.

After you enter the starting location, the screen will be initialized to high-resolution graphics mode with a single point at the center. This is the cursor point. At the bottom of the screen will appear a BYTES USED display, and the words ON OFF with OFF printed in inverse video.

The program recognizes the following keyboard inputs:

U	Move up one space
D	Move down one space
R	Move right one space
L	Move left one space
P	Invert point status
Control-W	Start over ("wipe")
Control-F	Shape is finished

Gary D. Dawkins, The Computer Shop/Vanguard Systems Corp., 6812 San Pedro Ave., San Antonio, TX 78216.

```
    0 REM  FROM THE COMPUTER SHOP
    1 REM  SAN ANTONIO, TEXAS
    2 REM   HIRES SHAPE VECTOR
    3 REM   TABLE ASSEMBLER PROGRAM.
    4 REM   APPLE HIRES MACH. LAN.
    5 REM   ROUTINES MUST BE RESIDENT
    6 REM   SET HIMEM:8192
    7 REM   LOMEM MUST BE 2048 (STAN-
    8 REM   DARD) FOR THIS PROGRAM TO
    9 REM   WORK.
   15 DIM D(3),LLL(4),HEX$(16),LOC$(4),H(4)
   16 HEX$="0123456789ABCDEF":INIT=3072:CLEAR=3086:PLOT=3780:SHAPE=3805
   30 POKE -16368.0: TEXT : CALL -936: PRINT : PRINT : PRINT "PLEASE ENTER THE LOCATION IN
      MEMORY YOU": PRINT "WISH TO START T HE VECTOR TABLE AT (IN"
   31 INPUT "HEX):",LOC$: CALL INIT:PFLAG=0:YCO=79:XCO=139: POKE 812,255: GOSUB 800
   35 FOR I=1 TO 4:H(I)= ASC(LOC$(I))-176: IF H(I)>9 THEN H(I)=H(I)-7: NEXT I
   36 LOC=0: FOR I=1 TO 3:LOC=LOC+H(I):LOC=LOC*16: NEXT I:LOC=LOC+H(4):LOC1=LOC: CALL -936
   38 VTAB 22: PRINT "BYTES USED:";
   40 VTAB 22: TAB 34: PRINT "ON ";: POKE 50,63: PRINT "OFF": POKE 50,255
   50 POKE -16368,0
   60 KEY= PEEK (-16384): IF KEY>127 THEN 100
   70 GOTO 60
  100 IF KEY=213 THEN 200: IF KEY=196 THEN 250: IF KEY=210 THEN 300: IF
      KEY=204 THEN 350: IF KEY=208 THEN 400: IF KEY=134 THEN 500: IF KEY=151 THEN 30
  150 GOTO 50
  200 YCO=OLDY-1:A=4: IF PFLAG=0 THEN A=0: GOSUB 1000: GOSUB 800: GOTO 50
  250 YCO=OLDY+1:A=6: IF PFLAG=0 THEN A=2: GOSUB 1000: GOSUB 800: GOTO 50
  300 XCO=OLDX+1:A=5: IF PFLAG=0 THEN A=1: GOSUB 1000: GOSUB 800: GOTO 50
  350 XCO=OLDX-1:A=7: IF PFLAG=0 THEN A=3: GOSUB 1000: GOSUB 800: GOTO 50
  400 PFLAG= NOT PFLAG: IF PFLAG=0 THEN 410
  402 VTAB 22: TAB 34: POKE 50,63: PRINT "ON";: POKE 50,255: PRINT " OFF": GOTO 50
  410 VTAB 22: TAB 34: POKE 50,255: PRINT "ON ";: POKE 50,63: PRINT "OFF": POKE 50,255: GOTO 50
  500 GOTO 1080
  800 IF YCO<0 THEN YCO=0: IF YCO>159 THEN YCO=159: IF XCO<0 THEN XCO=0: IF XCO>279 THEN XCO=279
  805 IF PFLAG=0 THEN 820: POKE 812,255
  810 POKE 802,YCO: POKE 801,XCO/256: POKE 800,XCO MOD 256: CALL PLOT: GOTO 840
  820 POKE 812,0: POKE 802,OLDY: POKE 801,OLDX/256: POKE 800,OLDX MOD 256: CALL PLOT
  830 POKE 812,255: POKE 802,YCO: POKE 801,XCO/256: POKE 800,XCO MOD 256: CALL PLOT
  840 OLDX=XCO:OLDY=YCO: RETURN
 1000 F=0:D=D+1:D(D)=A: IF D<3 THEN RETURN
 1005 IF D(3)#0 THEN 1010
 1006 F=1: GOTO 1030
 1010 IF D(3)<4 THEN 1030
 1020 Q=D(3):D(3)=0
 1030 Z=D(1)+D(2)*8+D(3)*64
 1035 BYTE=BYTE+1: VTAB 22: TAB 12: PRINT BYTE
 1040 POKE LOC,Z:LOC=LOC+1: IF F=1 THEN 1050
 1045 IF Q=0 THEN 1070
 1050 IF D(2)<>0 THEN 1060
 1055 D(1)=0:D(2)=Q:D(3)=0:Q=0:D=2: RETURN
 1060 D(1)=Q:D(2)=0:D(3)=0:Q=0:D=1: RETURN
 1070 FOR I=1 TO 3:D(I)=0: NEXT I:D=0: RETURN
 1080 REM  *** CTRL-F GOES TO HERE.
 1090 Z=D(1)+D(2)*8+D(3)*64: POKE LOC,Z
 1100 IF Z=0 THEN 1120
 1110 LOC=LOC+1: POKE LOC,0
 1120 PRINT "VECTOR TABLE: FROM ";LOC1;" TO ";LOC
 1121 PRINT "HEX: FROM ";LOC$;" TO ";
 1122 FOR I=0 TO 4:LLL(I)=0: NEXT I
 1123 FOR I=3 TO 0 STEP -1
 1125 IF LOC<16 ^ I THEN 1140
 1130 LLL(I)=LLL(I)+1:LOC=LOC-16 ^ I: GOTO 1125
 1140 NEXT I:LOC$=""
 1150 FOR I=3 TO 0 STEP -1: GOSUB 1160: NEXT I: PRINT : GOTO 1300
 1160 FOR J=0 TO 15: IF LLL(I)=J THEN PRINT HEX$(J+1,J+1);: NEXT J: RETURN
 1300 POKE 804,LOC1 MOD 256: POKE 805,LOC1/256: POKE 812,255: POKE 806,1: POKE 807,0
 1360 CALL INIT: CALL PLOT: CALL SHAPE: GOTO 32001
32000 PRINT ( PEEK (202)+ PEEK (203)*256)-( PEEK (204)+ PEEK (205)*256)
32001 END
```

A shape is drawn on the screen using the U, D, R, and L keys to move the cursor point. A point will be left at the previous position if ON is printed in inverse video. A point will not be left if OFF is printed in inverse video. The status may be reversed with the P key (OFF to ON and vice versa.) Remember that moving the point without plotting a point is legal data and will be assembled in the vector table. The BYTES USED will indicate how many bytes the vector table currently occupies.

If you wish to start the program over, enter control-W, and the program will again ask you at what location you wish to start the vector table. For multiple shapes, you may start at different locations or, to correct a shape, you may start at the original location.

Note that there are no editing facilities. A shape or portion thereof may be 'erased' by going over it with the point OFF, but this is clumsy and can lead to errors. It is suggested that the user initially draw the shape on graph paper, and work out the sequence of steps needed to draw it.

When you have finished entering the shape, enter control-F and the starting and ending locations of the vector table (in base 10 and base 16) will be displayed. To confirm your shape, the program will clear the screen and recreate it with the SHAPE subroutine from the Apple high-resolution graphics routines incorporated in this program. If this is not the shape you entered, one of two things has happened:

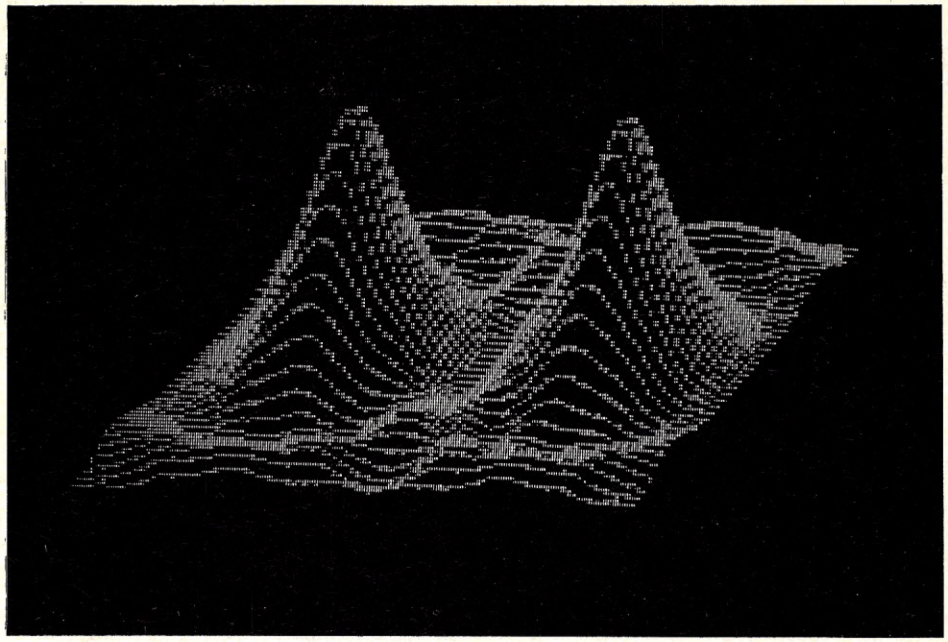
1. You made two or more moves up with the point OFF. This is not a fault of this program but intrinsic in the high-resolution routines as supplied by Apple. Plan your shape so that you do not have to make more than one consecutive move up without plotting.

2. Remember that the cursor point is never in the vector table. Therefore, the last point on the screen is not in memory. Always move one extra point (in any direction) before hitting control-F.

Some Notes for Those Unfamiliar with High-Resolution Graphics

This program incorporates the Apple HIRES graphics routines which reside in the range COO to FFF. These routines are described in detail in the current Apple user's manual. This program uses INIT, CLEAR, PLOT, and SHAPE.

The shape you create may be recreated if it is stored. You may record the shape by hitting RESET, entering the starting address (in hex), decimal point, ending address (in hex), W, and pressing RECORD on your tape player,



followed by RETURN on the computer. A shape starting at 4001 and ending at 403A would be recorded with this command: 4001.403AW. To read the shape back into the Apple, position the tape, and type in the starting and ending addresses as before, with an R in place of the W. Example: 4001.403AR. You may then pass the starting location of the vector table (the shape) by passing the starting address (4001 or 16385 decimal in this case), the color, size and rotation factors, and invoking the SHAPE routine. Familiarity with these routines is recommended. Good luck and good plotting. ■

Three noted game designers speak out on things that go beep, bonk, sprong, zap and zing on your TV set.

Video Games: Three Perspectives

David H. Ahl

Nolan Bushnell

Nolan Bushnell, 35, founded Atari in a garage in 1972 with $250.00. He invented Pong, which revolutionized the commercial arcade-game market. Since that time, Atari has been at the forefront of the video-game industry which, while still in its infancy, is now a $400 to $500-million-a-year business. In October 1976, Bushnell sold his company to Warner Communications for $28 million. He remains as Chairman of the Board and Chief Executive Officer.

I talked with Nolan about the market, his company, and their customers. Later Christopher Cerf of Children's Television Workshop/Sesame Street fame, joined the conversation. Here are some portions of our conversation.

Ahl: What kind of people are buying the more expensive, top of the line, video computer systems?
Bushnell: We're finding out that it's turning out to be everybody. We at first thought of ourselves as a carriage-trade item, but it's turned out to be a family toy or a family investment. Now, instead of popping $500-600 for a pool table, they're seeing video games as something the whole family can do together.
Ahl: Are users mostly people with kids?
Bushnell: The biggest buyers are a mix of fathers between 35 and 45 with boys, and young adults without children.
Ahl: There was some discussion shortly after arcade games were introduced that people tired of them and the manufacturers had to continue to introduce new variations. Does the same thing happen with the home games?
Bushnell: Well, first of all, let me say that it's not axiomatic that it happens in the coin-op games. There are the "hit records" and there are the classics in coin-op. Right now, *Breakout* has every appearance of being a classic like some of the sprint driving games in that they appear to be endless in their appeal. Then, there are the dime novels that seem to have a Roman-candle kind of existence. I think the same thing's going to happen with the video games — there will be the classics that seem to last forever and ever, and there will be the Roman candles that quickly die out. I think one of the reasons the video computer system has been successful is that people see it as a way to hedge their bets.
Ahl: Do you see the games competing with television? Are people watching TV less and playing games more?
Bushnell: I hope so.
Ahl: I do too.

There are ways to hand-lead Mr. and Mrs. America into the computing process without them knowing it's happening. When it comes to the computer, the only thing we have to fear is fear itself.

Bushnell: I guess we should talk about giving us a Neilsen rating.
Ahl: Given a relatively broad market, is your market strategy aimed at the consumer outlets rather than some kind of specialty store.
Bushnell: Yes. Bloomingdale's, Macy's and Sears are our outlets.
Al: But now as these games become more sophisticated with number key pad inputs, programmable games and whatever, aren't you going to need a somewhat more knowledgeable sales person to demonstrate them?
Bushnell: No. Absolutely not. If that's true, we've designed the product poorly. In order for it to be mass marketed, Mr. and Mrs. America are going to have to be able to use it and operate it. There are ways to hand-lead Mr. and Mrs. America into the computing process without them knowing it's happening. When it comes to the computer, the only thing we have to fear is fear itself. So, our product will be computers, and the consumer will be programming without ever knowing it. If we can't build a product that way, then we've missed our market and we'll end up selling only to the type of people who subscribe to your magazine.
Ahl: I was just up at Bloomingdale's. Anybody could walk up to *Stunt Cycle,* press a button, make a few mistakes and learn what had to be done to play. The same was true of video pinball, and probably with most of the cartridge games. On the other hand, one of the top of the line video systems, *VideoBrain,* was there with no instructions. People were just hitting buttons not knowing what to do. It was probably something that you could learn in fifteen minutes or less, but it wasn't very effective without a salesperson to demonstrate it.
Bushnell: A poorly designed product.
Ahl: That's a bit harsh.
Bushnell: No, I mean by requiring an expert to demonstrate the product, you're saying there's a small marketplace.
Ahl: From a product standpoint, what are you aiming at? You now have numeric key pads for your video computer system, what's the next step?
Bushnell: We're expanding. Using the numeric pads, we have a game called *Matching,* which is simply the old TV game of *Concentration* and some others, but the video game system has a major drawback — it has virtually no memory. The minute you postulate that if you want to do BASIC or something like that, then you're talking about a different system. However, since the video computer system is currently all sold out this year, we're holding off putting any more on our plate. Next year, it will be a new story for us. We believe that maybe Mr. and Mrs. America will be ready for a more challenging task.
Ahl: Meaning programming? Or a programmable unit of some sort?
Bushnell: Yes.

Nolan Bushnell of Atari

Nolan Bushnell

Ahl: In talking about programmable games, or moving more in the programmable direction, there are the tank war games where you set up your own pattern of mazes. That would be one level of programming. Another level of programming would be writing a game from scratch in a language like BASIC.

Bushnell: Whether you're talking about varying ball speeds or just working with some of the parameters, I believe there are ways to sort of back-door Mr. and Mrs. America into programming without them calling it that. Once they call it that, they're going to get scared of it. What we really want is for them to end up doing the task and then be hit over the head with "Hey, for the last month and a half you've been programming," and they say, "Oh no, not me. I'm not very good with math." All of a sudden they'll scratch their head and say, "Well, maybe I'm better at numbers than I thought I was." We'll also find the guy who gets the system, unpacks it, goes through the recipe and says to his wife. "Hey, I just programmed the computer!" when really he's just followed a recipe. So we can really bring what I call the "psuedo hobbyist" into the market as well. What we really want is the mass market because that will enable us to attractively price our systems. We'll have major computer systems for half or a quarter of what they currently cost.

Ahl: Do you see future video computer systems having communication capabilities?

Bushnell: Absolutely. Don't forget Warner Communications, one of the major cable companies, is experimenting with QUBE in Columbus. You've got a 2-way, 60 megahertz data channel so you can pump a lot of stuff up and down that cable. Just stop and think about the Sesame Street kinds of things made interactive with the viewers! Just imagine Johnny sitting in front of his set and some Sesame Street character says, "OK, all the kids that think it's 'A' push the A button," and all of a sudden it says, "240 million kids were right and 320 million kids were wrong."

Chris Cerf: We wouldn't be quite that blunt about it, but that's the right idea.

Bushnell: But that's the kind of nutsiness thing we can do.

Cerf: The other thing that intrigues us at CTW about this is the opportunity for people who are funny writers, or creative people working with kids, to be involved in this kind of technology — one which they are now scared to death of. But, in fact, it's not so scary once you know even a little about it.

Bushnell: Well, two years ago we diffused the fear that people had of connecting something to the TV. In the next two or three years we have to diffuse the fear of programming as a word, and the fear of computers per se. I think that we can actually do that.

It's not math majors they hire for programming, it's music majors, because the rigor is more important than the mathematics.

Cerf: I don't think people now believe computers are simple and easy to manage. For years I thought it was hopeless, but now that I have one I find it's not scary at all.

Bushnell: It's anti-complex which a lot of people don't understand. I mean, to take a problem that you understand and to back through and break it down to its simplest denominators is somehow an easier process than taking a series of simple concepts and building up to the complex end. IBM has known for years and years that it's not math majors they hire for programming, it's music majors, because the rigor is more important than the mathematics.

Ahl: By the way, other than working at Sesame Street, Chris is also a computer and arcade game freak to an extent that I've never known before.

Cerf: I didn't realize I'd gone that far or it's so obvious.

Ahl: I asked Nolan before whether people get tired of games. Do you find yourself getting tired of some of them?

Cerf: I think so, but there are so many others, it's not a problem. For example, on the Apple I have now, I put *Hammurabi* up first because it's a game I always wanted. Having won it a few times, I kind of lost interest, but you don't if you can keep refining it. I find that a tremendous challenge.

Ahl: What do you both think is the benefit to people playing all these games other than just having fun?

Bushnell: There are so many. Games in and of themselves always pose this question because they are obviously not overly productive. There's the huge spirit that really gravitates you toward a good game, and there's the training of your mind and your thought processes to be a better human being by using some of the skills that we survived with hundreds and hundreds of years ago as a genus homo sapien. We know because of our soft non-survival life that we still have these facilities which need to be exercised. I don't know. You can go into all kinds of esoteric philosophy.

Ahl: I don't mean to necessarily do that. Chris and I have discussed the possible benefits of the use of games with young children — just the simple things like hand and eye coordination and development of motor skills, that sort of thing.

Cerf: I think there's no question that the games do that, just as the whole idea of Sesame Street is to make a game out of learning. It's almost subversive education in a way. But it seems to me that a lot of the education programs I see on computers, are to some degree technologically much more brilliant than what comes across. For example, a very involved program puts a fancy formatted 2 x 6 on a screen and tells you whether you're right or wrong if you get 12. It could be less sophisticated, but a really super game if it was simply personalized to the student.

Bushnell: Along that line, I'm convinced that our *Blackjack* cartridge is a much better math teaching tool than our basic math cartridge.

Ahl: And to teach logic, there's nothing that compares with *Mastermind.*

Bushnell: Precisely. Inclusive sets, exclusive sets. I really believe it's going to be a necessary skill in the future for man to deal with an ever increasingly complex life to unburden himself from pedestrian mental chores so he can concentrate on many other things, just

Christopher Cerf

as surely as the telephone has made it possible for man not to have to run two blocks every time he wants to communicate something [he was healthier then - Ed.]. I'm sure there are going to be a few people 100 years from now who can't write a program, but those poor people will probably be on welfare. You can choose where you want to be, you know. The person now who is essentially illiterate has a hard time being a member of society.
Ahl: Which maybe says that Chris and I should be even more concerned with education of youngsters today so that in twenty years people can better cope with increasing change. Do you think industry has a role, too?
Bushnell: In the early days of *Pong*, one of the visions of the corporation I was going to have was the educational products division basically using this cheap display computer technology that I've evolved to do an inexpensive "talking typewriter" [a device developed by O.K. Moore at the University of Pittsburgh Learning R & D Center - Ed.]. This idea was sort of laid fallow because the games hit so hard. It was like holding on to a tiger for dear life, but we now have the ability to do a talking typewriter. We're going to put the fear into the hearts of the parents of America that they're holding back their children.
Ahl: Is the talking typewriter a likely product from Atari in the future?
Bushnell: Not as you know it. But we're working on all sorts of things.
Ahl: We'll be waiting with bated breath.

In a video game, the screen has a life of its own independent of, but related to the players. I think the feeling is of two people interacting with each other in this "other" world on the screen.

Bernie De koven

Bernie De Koven is a noted designer of games of strategy, author of The Well-Played Game, contributing editor to Games magazine and Simulation/Gaming, and founder of the Games Preserve. A non-profit educational corporation, the Games Preserve is a game-playing co-op where one can play flying rings, ping pong, pool, puzzles, board games and more. It's located on a 25-acre farm in the rolling hills of Eastern Pennsylvania. For information write The Games Preserve, RD-1355, Fleetwood, PA. 19522. (215) 987-3456.

Ahl: Bernie, you've been mostly involved with what you might call manual games or games of strategy, but not electronic games. So, how do you see the electronic games fitting in with the overall world of games?
De Koven: Well, I see it as a completely different medium. The kind of interaction that takes place when two people are playing while watching a TV screen is very different. There is an indirect kind of communication. It seems to put them in another space that they enter together but yet they're very separate from each other. In a board game, nothing happens unless you make it happen. The board isn't animated. Whereas in a video game, the screen has a life of its own, independent of, but related to the players. I think the feeling is of two people interacting with each other in this "other" world on the screen. I'm trying to figure out what the social interaction is. I think that the medium has just the most incredible potential of any that I've seen. Even now, most of the games you see really aren't utilizing the medium. For instance, the paddle used in *Breakout* could be an oscillating cylinder. The possibilities are limitless and they need to be explored.
Ahl: Do you think it's important that people have the ability to alter a game or write their own versions?
De Koven: I think it's important, but I think it's difficult. If you give people too many alternatives, they get very confused. A game falls apart when people keep changing the rules and there's no focus. I think you have to either limit the kinds of choices that people have or provide them with some sort of educational system so they can learn which choice to make because each change affects the entire game. Handicapping systems are very important. In the Atari games, for example, you can have a novice or an expert rating for each player which allows you to handicap one player slightly.
Ahl: Should there be continuous levels of handicapping?
De Koven: Definitely. I like to play a well-matched game, whether I'm playing with an adult or a child. Just to be able to vary the width of the ball or the paddle is not enough for me. Maybe you could even use the electronics to automatically adjust the difficulty level within certain boundaries.
Ahl: You sound very excited about this medium. Are you planning to design some electronic or video games?
De Koven: Yes, definitely. I want to explore and work with as many different types of systems as possible. The field is amazingly exciting.

Sid Sackson and Bernie De Koven at the controls of a video game.

Sid Sackson

Sid Sackson is actively involved in the game industry as an inventor, collector, author and contributing Editor to Games *magazine. His published games included The Winning Ticket, Acquire and Sleuth. His books — A Gamut of Games, Beyond Tic Tac Toe, Beyond Competition and Beyond Solitude.*

Ahl: Sid, what new possibilities do you see with electronic games compared to board games?
Sackson: One thing electronic games do is allow a single player to play a competitive game without needing a human opponent. Another good use of the computer is where it acts as the moderator in a game between humans. Take, for instance, war games where there's secret information or unknown partial information — I think that's a field that needs to be investigated.
Ahl: Along with the positive aspects, do you see any negative ones?
Sackson: One limitation comes to mind: I still think there's something nice about handling the pieces of a game and putting them down in place. I don't think that electronics can replace that aspect of board games.
Ahl: Or give you a tactile sense?
Sackson: Right, like say with *Go* where you have the tactile sense of holding the stones and the aural sense of clicking them in place. I hope we never lose that.
Ahl: I hope not either, but I've learned never to underestimate the possibilities of electronic technology. ■

Profile of a Smart Electronic Game...

Atari Video Pinball

David H. Ahl

Atari Video Pinball is what has come to be known as a dedicated video game. By dedicated we mean that it is not programmable by the user and the 28 games that it plays are all built into the console.There is no plug-in ROM or cartridge containing other games. Essentially it's an extension of the Pong type of games, but at the moment it is unique. The other manufacturers of electronic games such as Coleco, APF, Unisonic, etc., do not make an equivalent to the electronic pinball game.

Hooking it up to your TV is simplicity itself. Like all video games it comes with a TV/game switchbox that has two sets of input connectors, one for the game and one for a TV antenna. It has a short 300-ohm output wire that hooks up to the antenna terminals of the TV set and a large slide switch on the side that can either be set to the game position or to the TV antenna position. It has a piece of very sticky tape on the back so that you can tape it to the back of the TV set and leave it in position. Atari recommends getting a TV/game switchbox for each TV in your house so that you can easily transport the game around from one TV to another without having to unhook the antenna terminals each time. I find this a worthwhile recommendation and in fact have so equipped all the TV sets in our house.

The game operates on Channel 3 which, if your area is like ours, is convenient because Cable Television also is on Channel 3. What this means, of course, is that you never have to change your TV channel selector. The game is powered by either six "C" batteries or a nine-volt battery eliminator. Obviously if you elect to use "C" batteries your best bet is the long-life alkaline or mercury batteries.

The unit itself measures 7½ by 13½ by 3½ inches high. At each side of the unit are buttons that control the flippers and on the top of the unit is a large knob at the right side that controls the paddle. There's one paddle at the bottom of the screen which moves back and forth across the screen. There are four buttons on the console: one switches the power on and off; another is a reset button which resets each game and sets the score to zero. There is a game-select button that cycles through the seven major games in order, and an option button that cycles through the four options of each of the games. When you turn the unit on, it automatically switches to game number one which is flipper pinball (variation one). If you want to play breakout, for example, that is game number seven and you would hit the select switch seven times to get to game seven. If you wanted to play the fourth option of breakout you would then hit the option switch three times to get to the last option of breakout.

A fifth, larger button on the console controls the ball serve; this is what you hit each time you want a new ball in any of the games served up. We tried the game on both a black and white and color TV set and found that the brilliant colors provide much more excitement in the play of the game. As your ball hits one flipper or bar or box it changes color, which provides an added element of fun to the play.

The first two games are flipper pinball one and paddle pinball one. In flipper pinball you use the flipper buttons on either side of the console to move the flippers at the bottom of the play field. The flippers remain in the upward position only momentarily when you press the flipper buttons. In other words, you can't leave them closed so the ball doesn't slip through.

In paddle pinball, you use the paddle-control knob to move the paddle horizontally back and forth at the bottom of the play field and hit the ball as it comes down to the bottom of the screen back up into play. The game automatically serves the ball when you press the ball-serve button. The object of the game, of course, is to direct the ball to the bars and blocks on the playing field and to get it to bounce around and hit them as much as possible. Starting out, there are four yellow bars at the top of the play field. Each time your ball hits any of these bars, you score 200 points. The first time a bar is hit, its color changes to red; after that, the red and yellow color alternate after each hit. Also, the ball speed increases after it hits any one of these bars. There's a large block in more or less the center of the playing field with four different rectangles in it. They change color from red to green when the ball hits them. The ball will actually bounce around inside the block seemingly at random, although obviously, it's not random. You score 100 points each time the ball bounces off an inside wall. There are also six green side blocks. When the ball hits any one of these blocks you get 100 points and the block changes color to blue. When you hit the blue block you score 200 points and the blue changes to red. Hit the red block and you score 400 points and the block disappears. Eventually all the side blocks disappear but then will automatically reappear when you change all of the four top bars of the playing field to the same color (either red or yellow). When you accomplish this you get a bonus of 1500 points. The paddle in paddle pinball will shrink to half its original size when the six side blocks disappear and reappear. As soon as you miss the ball the paddle returns to the full size on the next serve.

In flipper pinball two and paddle pinball two, the playing action is exactly the same as it is in the first game. The only difference is the scoring and the playing field. In this playing field, there is a white bar at the top two small yellow side bars, two side boxes, a center hoop, sort of like a

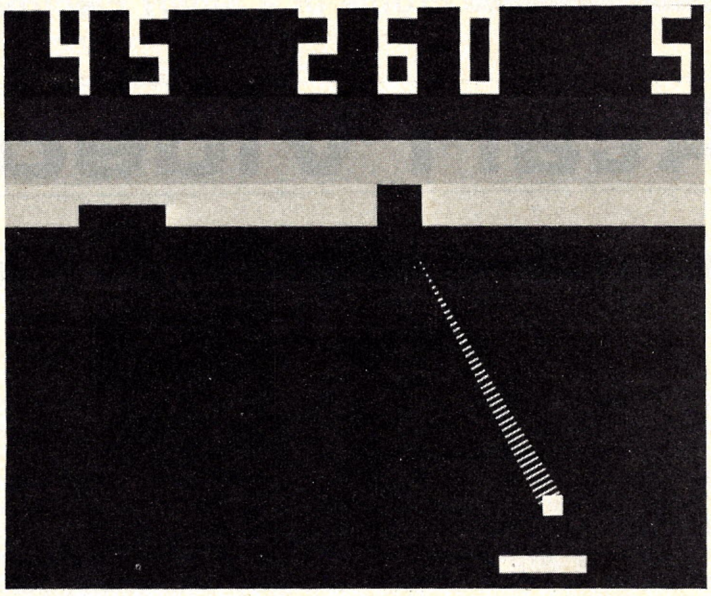
Pinball Continued...

basketball net and eight boxes at the bottom of the screen. Personally, I like the playfield on the pinball two games but there are obviously people who like each one. Typical scores for beginners playing with all seven balls are 20 to 30,000 points. A more proficient player can get up into the 100,000 point range and expert players can get well over 200,000 points.

The four playing options allow you to adjust to different playing-skill levels. Option one allows both flippers to move simultaneously and gives you seven balls. Option two also allows simultaneously flipper movement but only five balls. Option three only allows one flipper to move at a time; you have five balls. Option four also allows only one flipper to move at a time and gives you three balls. The paddle options are roughly the same. On the more difficult options, the paddle is half the size of the easier options.

The two rebound games, which might be thought of as basketball, have as their object to catch the dropping ball on the paddle and then using the paddle control to move the paddle back and forth, dribbling the ball in a sense. Then when you get into the right position, hit the flipper button to shoot the ball up and through the basketball net. The difference between rebound one and rebound two is that rebound one gives you one basketball net to shoot for; rebound two gives you three. When you make a point, the net goes from the left side of the court over to the right side and alternates back and forth as you make points. Your score is shown at the top left of the screen. The number of remaining balls, as with all of these games, is shown at the top right of the screen. If it is a consecutive game of rebound, your previous score is stored to the right of your present score, more or less in the center of the screen. That way two competing players can compare scores. I haven't played rebound that much to become a really good player but it seems as though scores in the twenties and thirties are about normal.

On a recent visit to Atari in Sunnyvale, I ventured into the "Game Room," a room containing most of Atari's coin-operated games. Atari employees can play games free during breaks and before and after work. There I met a person who I can only describe as a Breakout Junkie. His goal was to break away both walls using only one ball. While I watched, he consistently broke away both walls with three or four balls. Whew!



Undoubtedly, the most popular game on Atari Video Pinball is one called breakout. The object of the game is to remove a wall of bricks one at a time. For your sledge hammer you have the ball. The wall consists of three color bands of bricks — yellow at the bottom, green at the center, and red at the top. When you strike a brick with the ball the brick disappears and the ball bounces back to the paddle. The number of points you score for each brick depends on its color. There are six rows of bricks with three different colors. Yellow bricks at the bottom score one point, green bricks in the center two rows score four points and the top two rows of red bricks score seven points apiece. If you completely remove the wall from the playing field you will have scored 432 points. The wall then automatically sets up a second time. The maximum score if you remove the wall completely two times is 864 points. After many many plays of the game I have never gotten up to the point of removing one full wall; I've gotten very close but always have wound up with five or six bricks remaining. In other words, my scores were up in the upper 300's and low 400's. When you strike a brick in the second green row or any red row, the ball speeds up. Also, the angle at which the ball bounces off the paddle increases up to the thirteenth hit, but decreases again as the ball speeds up, thus making it somewhat easier to control. You'll find on the 9th thru 12th hit that the ball is bouncing off the paddle at some very extreme angles, thus making it very difficult to control. Any time you miss a ball and the ball is served again, it returns to its original speed and the paddle returns to original size. Your score is in the upper left corner in the play field and the number of remaining balls is in the right corner. As in rebound, if you are playing a consecutive game of breakout, the previous game score will be stored on the screen to the right of the new score. That way, two competing players can compare scores.

The four Breakout options are: option one, paddle regular size, seven balls; option two, paddle regular size, five balls; option three, paddle half-size, five balls; option four, paddle half-size, three balls. Frankly, options one and two are more than enough challenge for most people and at this point at least, I find option three and four rather frustrating. Perhaps if I were playing eight hours a day for weeks on end, I might get up to proficiency level to handle those options.

The instruction manual is very complete, even going so far as to give a troubleshooting check-list in the last pages. The cause and remedy for the various trouble symptoms are somewhat simplistic. On the other hand, they do give the ignorant customer a starting point for debugging his system.

Atari Video Pinball is also marketed by Sears as Tele-Games, Pinball/Breakaway. Pricing ranges between $64.95 and $89.95. Clearly, as with all of these electronic games, it pays to shop around. Some of the stores in this area sold their demonstrators at greatly reduced prices after Christmas. In general, on these products I wouldn't recommend buying a demonstrator. They have taken very very rough abuse over a long period of time and frequently the knobs are loose and the switch contacts aren't in the best of shape. You may get one that is in good shape or be able to refurbish it yourself, but in general, I think the few extra dollars to get a factory-packaged new game are worth it. ■

A Creative Computing Equipment Profile...

Atari Video Computer System

David H. Ahl

This system is one of the most comprehensive programmable video games. It has a microprocessor chip — a 6505 to be exact — but it really doesn't matter very much what it is. You are not likely to be interfacing a unit like this to a computer or terminal or some other peripheral. So what really matters is how it performs as a self-contained unit.

Compared to the other programmable units on the market in late 1977 and early 1978, in particular those made by Coleco, RCA, and Fairchild, in our opinion the Atari is head and shoulders over the others. Three new units have been subsequently announced by Bally, VideoBrain and Coleco which are more competitive. After we've reviewed all of them individually, we'll give you some comparisons.

The Video Computer System hooks up to a TV set as do most of these games by taking a TV/game switchbox, hooking the Video Computer System to one side of it, your TV antenna to the other, and the output 300-ohm cable to the VHF input terminals on the TV set. This particular TV/game switchbox is nice in that it can accept either a 300-ohm input or a 75-ohm coax cable as are found on most CATV set ups. The TV/game switchbox contains a matching transformer for the 75-ohm input. Also a matching transformer for the game input since the output from the game is also 75 ohms and not 300 ohms. The unit is powered by a 9-volt AC power supply which is included with the system. While the instruction manual says nothing about ever unplugging the power supply once it is plugged in, we recommend for prolonged periods of inactivity, certainly if you're going to be away from the house on vacation for a long weekend, unplugging the power supply. It doesn't draw any more current than a doorbell transformer. On the other hand, we noticed after it had been plugged in for a month at a time it got a little bit warm. This unit uses the sound system on the TV set so you can adjust the volume control for either minimal levels or to overpowering levels if you like the sound of tanks rumbling around in your living room.

The dimensions of the unit are 23-1/2 inches by 13 by 4-1/2 inches high. It's made of black plastic with a simulated wood grain panel on the front. It comes with two joystick controls which have red firing buttons in the upper left hand corner of the control. The controls serve different functions in the different games. The joysticks are currently used with the tank games, space games, target games and chase games and hangman. The unit also comes with two paddle controls which are actually just rheostats or knobs which are used for the pong games, speedway games, black-jack, and math practice. A third type of controller which is a rheostat with no stops is used for the Indy 500 racing games. This controller does not come with the original unit and must be purchased separately when you buy the plug in ROM cartridge for the Indy 500 game. A fourth type of controller which also must be purchased separately is a numeric keypad with the digits 0 to 9 and two control characters (exactly the same as a Touch Tone phone). This is used with the Codebreaker and Hunt and Score cartridges.

There are six controls on the top of the console — an on/off power switch, a TV type switch (color or black and white), two controls which determine the difficulty level for the left and right side player. As a handicap for more experienced players, one side can be played in the A position and the other side in B. Details of the switch action are explained in the individual game booklets supplied with the game programs. There is a game select switch which cycles through the various games in each cartridge and next to it is the game reset switch which initializes the game and sets the score to zero. We probably would not have put these two switches next to each other because occasionally we hit the game select switch instead of the game reset switch and, if there are thirty or forty games on the cartridge, as some of them contain, it means you have to hit the game select switch forty more times to get back to where you were. In the center of the console is a what's called a "Game Program Slot". This is an opening for the ROM memory cartridge to be inserted. This cartridge

ATARI Continued ...

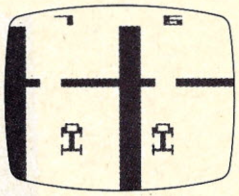
is slightly larger than a standard cassette tape. When the unit is completely set up with four or more controllers, the wire to the power supply, and another wire to the TV set, it certainly produces a rather large gaggle of wires lying around, in particular, six or more ten-foot cords. Clearly, this is not something that pleases your average housewife. On the other hand, short of wireless transmission we don't see any easy way around it.

As of this writing, there are sixteen cartridges available for the Atari Video Computer System.

Combat. This cartridge comes with the video computer system and plays three different types of combat games with tanks, bi-planes and jet fighters. There are five different playing fields, a tank open play field with no obstacles, an easy maze playfield with six obstacles, and a complex maze play field with fourteen obstacles. There are fourteen variations of tank games, ranging from ones where you simply shoot a straight missile at each other to shooting guided missiles with direct hits and/or billiard hits (that is where they bounce off the walls and obstacles). A guided missile is one that when it leaves your tank can be guided by your joy stick on a curved course to (hopefully) hit your enemy. There are five invisible tank games in which you and your opponent are invisible to each other except when a missile is fired or when a hit is made. In addition, the tanks become visible whenever they bump into a wall or a barrier.

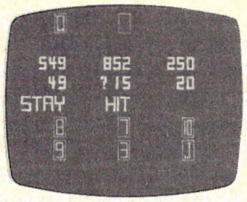
In the airplane games you have the options of playing with or without clouds, you may play with guided missiles, straight missiles or machine guns which are essentially the same as straight missiles although with a much lesser range. The jet fighter games are much the same as the bi-plane games except that the maneuverability of the fighter is controlled differently with the joystick. Also, they seem to go a little bit faster. All games end when either player scores 99 points or when game time runs out. I have never played a game that has gotten close to 99 points before the game time runs out.

Starship. In this you have four play fields providing seventeen different games. In one, you are the astronaut looking out of the window of your spacecraft, flying in two-dimensional, star-studded space. The object of the game is to use your force-field to draw your opponent's craft into laser range. He doesn't have power to resist but within a certain radius of your vehicle he has power to become invisible. You score points within a time limit for shooting his vehicle but deduct points if you collide with asteroids. You can test your perception in Warp Drive which gives the appearance of flying through three-dimensional space or you can guide your vessel in for a lunar landing on the moon, controlled by your opponent or a robot mechanism. We found the sound effects in this game added a great deal — eerie space noises and a resounding BLAM whenever anything gets hit.



StreetRacer™

Street racer. You and your opponent race through traffic against the clock controlling speed and steering; add points for passing cars safely, deduct points for collisions. Other games include slalom skiing, in which you ski down a slope through the slalom gates; Dodgem, in which you drive your car over bridges, but lose points by crashing; Scoop Ball in which you trap a ball as it zips down a track, score by passing it to the next racer; Number Cruncher in which you score points by running over passing numbers; Jet Shooter in which you fly your plane past oncoming fighters and try to shoot them down.

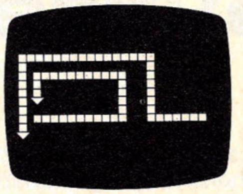


Blackjack

Blackjack, as its name implies, is a game for one, two or three players in which you play blackjack against the house using standard Las Vegas rules.

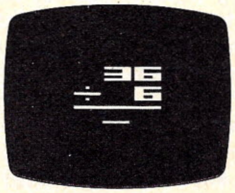
Video Olympics. These are 50 variations of the familiar pong games but what variations they are! You can build your opponent's frustration by volleying back and forth before deflecting the ball into his goal, you can have automatic or manual ball speed up. Games include pong, super pong, Foozpong, Quadrapong, soccer, hockey, volleyball and basketball. Options include, of course, the expert novel novice paddle size, a "flim-flam" option which lets you catch the ball and reposition before releasing it. Or one player can challenge the built-in robot player. Games are scored to 21 points.

In **Indy 500,** you have four play fields for fourteen games. You control the direction and speed of your car in a race to score more laps than your opponent within the time limit. You can choose complex or easy circuits. As your skill increases, try playing Ice Race where the racing surface is designed to be slippery or try Crash N'Score — an attempt to slam a car into a blinking target randomly placed on the screen or play Tag where players alternately try to tag each other.



Surround™

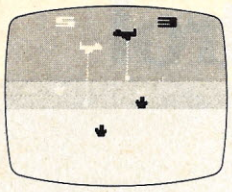
Surround. This cartridge provides fourteen games on three play fields. Use strategy and timing to force your opponent to steer his track through a track that has already been laid. If that sounds a bit complicated, it is. What you are each doing is laying down a track in a sort of maze type of arrangement and trying not to get trapped by your opponent. The track can wrap around off the screen to the left and then reappear on the right side, a rather interesting variation. Other variations include speeding it up, in which you hear and see tracks go from first to fifth gear or diagonal movement in which the race leaders blocks diagonally. Another option is to erase in which you can add intrigue by pushing your controller button so that no blocks appear until the button is released. This game can be played either against an opponent or against the computer itself. One nice variation on this cartridge is the TV graffiti option in which you can write a word or draw pictures on the TV screen in a free form mode. The sound effects on the Chase games are fantastic; whenever you hit a barrier you hear a big "SPRONG" with a reverb echo effect.



BasicMath™

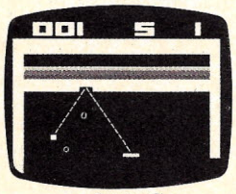
Basic Math. In this game you can multiply, add, subtract or divide picking a base number to work with in a series of ten problems. At the moment, this is a rather cumbersome game to play since the input is via the paddle controller. It seems to us that the numeric keypad controller would be more suitable for this game package. One nice touch is that when a correct answer is given, a little musical tune plays rather than just lighting "yes" or "that's right" on the screen.

ATARI Continued ...



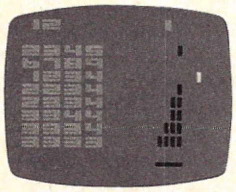
Air-Sea Battle™

Air/Seabattle. In Anti-Aircraft game you may fire an anti-aircraft gun at randomly flying jets and helicopters. In Torpedo you move guns along the bottom and fire torpedos at ships; in Polaris, you control the speed of ships and missiles while shooting at planes. Bomber is like missile but planes drop bombs on passing ships; in Polaris vs. Bomber one player is a plane and the other is a ship. There are also Shooting Gallery games in which you angle your gun and fire at clown, duck and rabbit targets which are darting across the screen.



Breakout™

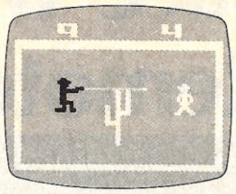
Breakout. Believe it or not, here are 48 versions of the popular arcade Breakout game. On the cartridge there's original Breakout, of course, along with 36 variations which introduce the player to the confusing world of gravity, time, funny paddles, and invisible bricks. We particularly liked the two-player variations which alternate between players on each shot rather than at the end of the game.



Codebreaker™

Codebreaker. Here are some old computer game favorites — six variations of Bagels, six of Mastermind, and eight of Nim. Need we say more?!

Hunt & Score. This is a video version of the TV quiz show "Concentration." Thirty numbers flash on the screen. Behind each one is a familiar object or a wild card. Match any two and score. The cartridge contains eight game versions for one and two players. We had a prototype of this game on loan for several weeks but no instructions; within hours my 7-year old had figured it out and was challenging (and beating!) the rest of the family as well as the whole neighborhood. Minnesota Fats move over!



Gunslinger

Outlaw. In the 12 Gunslinger variations you shoot it out with an opponent — sometimes with a cactus, wall, or stagecoach (stationary or moving) between you. Some variations permit bullets to ricochet or blast away the object between opponents. Others limit the number of bullets you have. Four more single-player variations allow you to shoot at a moving target. Warning: after a couple of glasses of wine, Chris Cerf and I went into fits of convulsive laughter playing this game and lost our ability to fire straight. This could happen to you!

Other games currently available but not reviewed here are **Space War, Home Run** and **Hangman.** Soon to be released are **Football** and **Basketball.**

Another nice touch with the Atari system is that every few seconds the playing field colors change. This prevents one particular phosphor from getting worn out. Even if one game is played repeatedly its pattern will not become etched into the screen. This is a real plus not found in any of the other video computer systems.

The Atari Video Computer System is also marketed by Sears as Tele-Games Video Arcade System. The price for this system is up to $230 in retail specialty shops. The Spring and Summer 1978 J.C. Penney catalog lists it at $164.95, about as low as I have seen it anywhere. Cartridges cost about $19.00 apiece. Clearly if you were to buy all fifteen cartridges currently available, at $19 apiece, and the extra controllers, you would wind up spending considerably more on cartridges and controllers than the Video Computer System cost in the first place. Initially, it is probably worth finding a retail store that has most or all of the cartridges on display and trying them out, selecting the two or three that seem to most strike your fancy and starting with that. If you then tire of them later on, you can always buy additional cartridges. (Personally, if this were my only system, I'd go with Video Olympics, Surround, Outlaw, and Breakout for starters along with Combat which comes with the system.)

All in all, the Atari Computer System appears to be well made, the games are designed with many hours of fun in mind, and, if your family is anything like mine, you'll find that they do indeed provide countless hours of enjoyment for children of all ages. ■

Interface with AC Remote Control

Debbie Pace

The microcomputer industry is rapidly bringing the size and cost of computers well within the realm of the average household. At the same time, microcomputers are becoming such powerful tools that the vast majority of us won't even begin to scratch the surface of our computer's capabilities.

For all the games we play, checkbooks we balance, programs we write to plan menus, etc., we are probably using our microcomputer only about five percent of the time. With all that power available and with the kind of dollar investment most people are making for their computer system, it seems there should be some way to make use of the computer the other 95 percent of the time.

The printed-circuit boards in front are the AC Controllers (Apple II at left, S-100 board at right); two Dual Channel Remote units are shown in the background

AC Remote Control

Mountain Hardware has one answer, with their Introl Remote Control system. A user can put his underused computer to work controlling appliances and electrical devices anywhere in his home or office. Very simply, the Introl system consists of an AC Controller for either Apple II or S-100 bus computers, an AC Interface Adaptor, and Dual-Channel AC Remote units located wherever control is to be established. The Introl system uses the existing 100VAC wiring to communicate between the computer and the remote devices, thus eliminating the need to run external wires.

How it Works

The AC Controller plugs directly into your computer bus and generates a coded signal which is impressed on the AC line. The AC Interface Adaptor serves to isolate the Controller and your computer from the 110V power while allowing the signals to pass. At the receiving end, Dual Channel AC Remote units may be plugged into any available wall socket. Each Remote can control two independent 500-watt devices. The Controller and Remote units must be tuned to the same frequency, around 50 KHz. This is easily accomplished through on-board, self-calibration circuitry. The Remote unit decodes the digital information signal from the AC line and turns devices plugged into it either on or off in response to the command it receives from the Controller. At 30 commands per second, the system is capable of rapidly activating a number of strategically placed devices. The Controller can also poll any channel and determine its present status. This bidirectional communication is very important for applications requiring error-free operation.

Each Controller can operate as many as 32 Dual Channel Remotes, for a total of 64 independent channels. This is certainly more than the average home application would require, but it leaves room for expansion as you think of new applications you would like to use your system for. Also, a small business or industrial operation would not find themselves limited by lack of available channels, thereby making the system versatile enough for all types of users.

An address is assigned to each Remote via jumpers and in every transmission, whether from Controller to Remote or vice versa, the address is checked for accuracy. Each channel in every Dual Channel AC Remote is constantly monitoring the signal transmitted by the Controller. Only when it receives a code corresponding to its address does the Remote execute the command. In fact, using the standard 8-bit UART data format, every transmission is checked for parity, framing and overrun errors, as well as the address information.

The Controller is capable of transmitting accurately up to half a mile or to the local transformer. At this point, it may occur to many of you to ask, "If I can transmit up to a half-mile, what happens if my neighbor has an Introl system too? Won't we be turning each other's devices on and off?" The answer is "yes" with an unequivocal "but" attached. Obviously, you will know very quickly if this situation occurs. If it does, all you need do is tune your Controller to a higher or lower frequency and align your Remote units to the new frequency. This will resolve the problem immediately. However, there may be circumstances under which this situation could be used advantageously. For example, such a system installed in an apartment building or large office complex would enable one guard watching a video screen to monitor every door and window from a central location.

Program Your Introl System in BASIC

The programs you write for your Introl system can be as simple or as complex as your application requires. Most users will be happy to know that they can program in either BASIC or assembly language. There is also an Introl BASIC available from Mountain Hardware designed to make program-

Debbie Pace, Mountain Hardware, Inc., 5523A Scotts Valley Dr., Scotts Valley, CA 95066.

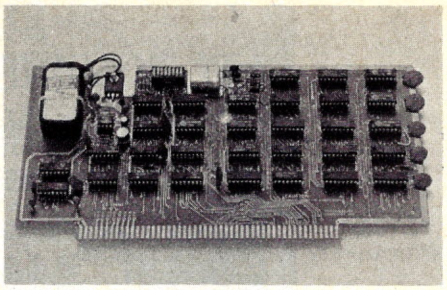
ming even easier while allowing a wide range of programming capabilities. It includes all the standard tiny BASIC functions as well as a special set of commands for control and clock functions. Some of the added statements allow you to get the status of a Remote (STAT$), turn devices on and off (SET), and check the Clock for print out in standard time format (TIME); for example, 10:33.

We've included a sample of a typical program written in Introl BASIC for remote control and clock applications. The program has two main functions. It continuously checks the status of all devices and activates the tv at a particular time so you don't miss your favorite program. Note: remark **statements precede the line number to which they refer.**

Expanding the System

There are many things you can do to make your Introl system even more versatile. By adding analog input you can remotely sense temperature, humidity, light variations; anything that can be read as a voltage or resistance. You could then use your computer to gather data, make judgements (based of course on your programming instructions), and then carry them out; for instance, sense inside and outside temperature and, based upon time of day and occupancy, determine whether room temperature should be increased or decreased and adjust the heater or air conditioner accordingly.

In many instances your system will need time and date information to be completely effective. Of course, this can be handled in software but why go to the time and effort when there are many clocks available on the market at reasonable cost? Mountain Hardware makes a clock that is especially well suited for use in remote control applications due to several rather unique features. It is crystal-controlled for accuracy and has an on-board, 9V rechargeable battery to keep the clock ticking away even during computer down times, intentional and otherwise! Also, it allows you to program time-related functions from 100-microsecond increments for periods as long as 100,000 days—that's 273 years! By adding a clock to your Introl system you could, for example, program lights to go on and off in a different sequence every day to simulate occupancy while you are away on vacation.



The 100,000-day Calendar/Clock Board for the S-100 bus consists mainly of a crystal-controlled clock, frequency dividers, circuits that combine the various frequencies to select the required time interval, and a NiCd battery for fail-safe operation.

Another rather intriguing way to expand the system is to interface Introl and a speech recognition unit such as the Heuristics Speechlab. One possible application for this combination would be part of a home security system. You could install solenoid locks on every door, to open only after you have given a preprogrammed sequence of words. The words are checked against the model you previously input into the Speechlab and if they don't match, the door remains locked!

More Applications

At the simple convenience level, you could program your system to bring the temperature in your house up to a

```
  5 REM ** SET UP PORT ADDRESSES FOR AC CONTROLLER
  6 REM ** AND CLOCK, THEN SET ERROR MAXIMUM TO 5
 10 CONTROL=4 : CLOCK =32 : ERRMAX=5
 15 REM ** MAKE TABLE OF CURRENT DEVICE STATUS **
 20 FOR I=0 TO 3: @(I)=STATUS(I): NEXT I
 25 REM ** SCAN DEVICES FOR CHANGE IN STATUS
 30 FOR I=0 TO 3: S=STATUS(I)
 35 REM ** IN CASE OF CHANGE PRINT NEW STATUS
 36 REM ** AND PUT NEW STATUS IN TABLE
 40 IF S    @(I) THEN GOSUB 100
 50 NEXT I
 60 REM ** CHECK TIME, OPTIONALLY CONTROL TV **
 70 IF TIME(1)=1900 THEN GOSUB 300
 80 IF TIME(1)=2000 THEN GOSUB 400
 85 REM ** KEEP CHECKING FOR STATUS CHANGES
 90 GOTO 30
100 REM ** SOMETHING CHANGED STATUS ** DISPLAY CHANGE**
110 PRINT #1, "DEVICE", I, "HAS CHANGED STATUS TO", STAT$(S)
120 @(I)=S:RETURN
300 REM ** TURN ON TV (DEVICE NUMBER 3) **.
310 SET 3,1: REM ** 1=ON, 0=OFF
320 PRINT "TIME FOR THE NEWS"
330 RETURN
400 REM ** TURN TV OFF (DEVICE NUMBER 3)**
410 SET 3,0: REM ** 0=OFF, 1=ON
420 RETURN
```

comfortable 68 degrees half an hour before you get up in the morning, then start the coffee, let the dog out and finally wake you up, either gently but persistently by sounding an alarm or if you prefer, by playing a rousing Sousa march on your stereo. You would of course program your system *not* to wake you up on weekends.

From here the next logical step would seem to be using Introl for energy management. Every electrical device in your home or building could be constantly monitored for power usage. You could even write a program to have your computer shut things down on a priority basis after a predetermined level of usage has been reached. At the same time, your system could monitor and control your solar heating system for optimum energy utilization and water your lawn at 2 A.M. while you are sound asleep (it's the best time, you know)! For those of you who are a bit on the forgetful side, you could install ultrasonic or infrared sensors in each room, then program your computer to turn off lights, tv, stereo, etc., five seconds after you leave the room.

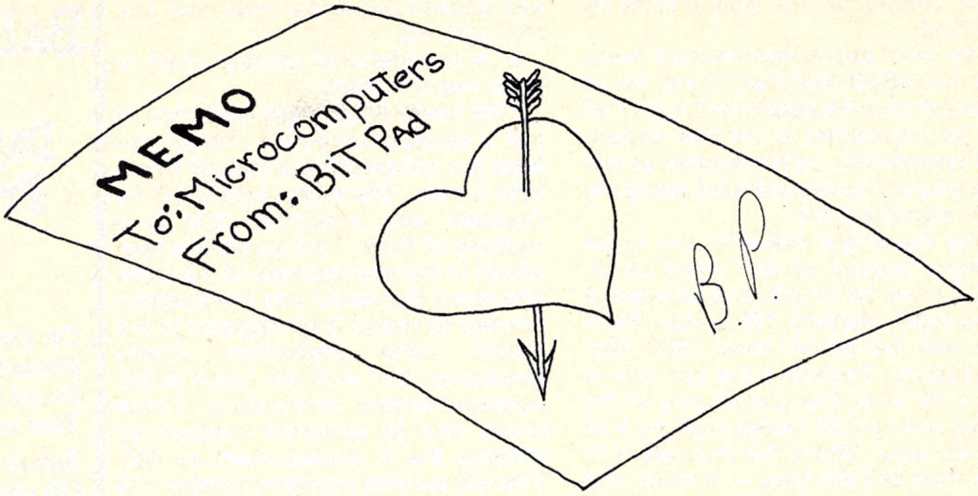
In the realm of home security, your computer system could become an intelligent burglar alarm by adding input sensors. You could program it to do one of several things depending on the conditions. If you are home, the system alerts you with a silent alarm. If you are away, the system dials a neighbor and/or the police and plays a prerecorded message or sounds a loud alarm to frighten the intruder, or. . . .you provide the options, and your system does the work.

A photographer could save both time and energy by setting up a fully automated darkroom. For those of you with a green thumb, an ideal place to introduce an intelligent remote-control system would be the greenhouse. With the addition of sensors, your computer could collect information on temperature, humidity and soil moisture and then activate devices to adjust these factors as needed.

The possibilities are endless. You're probably already thinking of several I've missed. So why not start using your computer to its full potential while you "turn on your life"!

The Bottom Line

Prices are: AC Controller (either Apple II or S-100), assembled and tested, $189; in kit form, $149. Dual Channel AC Remote units, assembled and tested, $149; in kit form, $99. 100,000 Day Clock, assembled and tested, $219; in kit form, $179. For more information, please address Mountain Hardware, Inc., 5523A Scotts Valley Drive, Scotts Valley, CA 95066 or phone (408) 438-4734. ■



Interfacing with a Bit Pad

Robert Davis

The formality of a keyboard versus the freedom of a pen is perhaps exaggerated and possibly a bit frivolous as represented in the above "Memo," but consider for a moment the task of entering the data into a computing system. The picture may well require 1000 bytes to describe it to the system. This would, of course, require a corresponding number of keystrokes plus many measurements.

Using a data tablet (read "Bit Pad") to enter the graphic portion of the data would require no more time or effort than the original drawing.

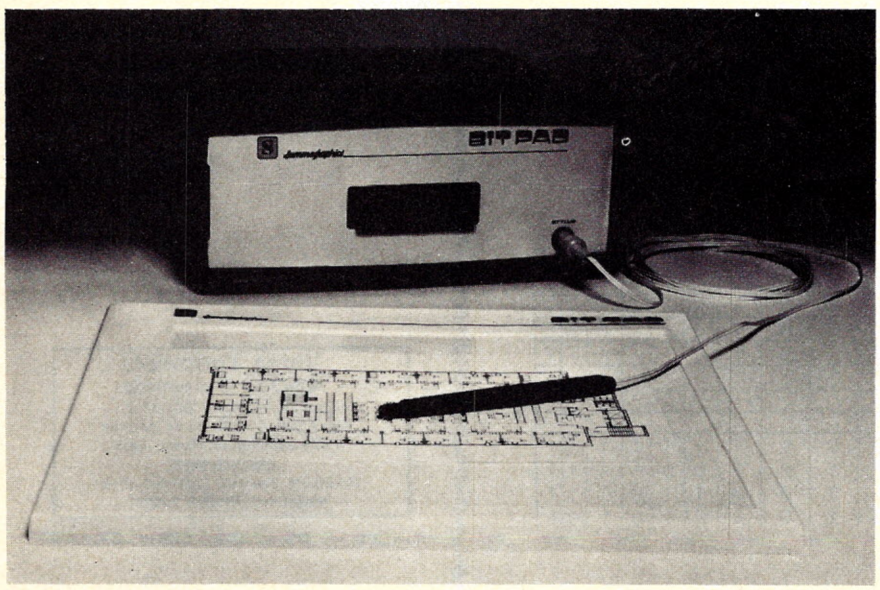
Robert Davis, Summagraphics Corp., 35 Brentwood Ave., Box 781, Fairfield, CT 06430.

Two rather straightforward routines will yield area and total perimeter of the shape being described.

These two attributes are particularly useful in civil engineering (land-fill requirement) and in architectural estimating (bills of material based on square footage or linear runs). All of this data is available on drawings. It is only necessary to be able to accurately describe them to a computing system.

Resolution

Summagraphic's Bit Pad provides an economical entry device that is easy to use, precise and stable. With two resolutions available, .005 inch or .1 millimeter, the user has either 2200 lines over the 11-inch active surface, or 2895 lines. The Bit Pad will update the computing system with new location data 64 times a second. It will track a stylus (pen) or a cursor on its surface.



The Bit Pad, with digitizer board, stylus, console cabinet (with controls for selecting the data-collection mode), and RS-232 output, is $555.

Cursor Option

A cursor, or mouse, looks like a hockey puck with a bombsight. It provides a more accurate crosshair to pinpoint the spot to be digitized. The data sent to the computing system is an X-Y coordinate pair, in the form of two 12-bit binary words, created on command from an internal timer or a switch inside the stylus. This switch closes on contact with the tablet, and may also be used to control the internal timer.

Menu

It may be that the data being traced and entered into the computing system requires a scale factor to further define it. These functions may be included on the surface of the tablet in the form of a menu.

Yes, it could read (using a luncheonette order form as an example) like this:

HAMBURGER	1	2	3	4	5	6	7	8	9
HOTDOG	1	2	3	4	5	6	7	8	9
FRENCH FRIES	1	2	3	4	5	6	7	8	9
COFFEE	1	2	3	4	5	6	7	8	9

These *menu* items would be entered into a system by touching the appropriate quantity with the stylus. Y displacement would indicate the item, and X displacement would determine

quantity. These items could have been scale factors for the drawings being digitized.

The most easily implemented menu is a full ASCII keyboard. First, let us establish that the data from the Bit Pad will be transferred to the host system and operated on, by that system, as two 12-bit binary words with bit 0 being the least significant bit.

The menu will reside in the lower lefthand corner of the tablet (X=O, Y=O). It will be 16 columns wide and 8 rows high, yielding 128 boxes. Each box will be 64x64 lines (.005 inch resolution). Therefore the box size is .32 x .32 inch. The overall size of the menu will be 2.56 inches high by 5.12 inches wide. When the ordinate Y is less than 512 (64 lines — 8 rows) and the abscissa X is less than 1024 (64 lines — 16 columns) the stylus is considered to be in the menu area.

When in this area, the computer will construct an 8-bit byte made up of the following:

MSB	*Parity*	*Your choice*
	Y bit 8	
	Y bit 7	
	Y bit 6	
	X bit 9	
	X bit 8	
	X bit 7	
LSB	X bit 6	

Menus may be much more complex. I am presently working on a menu that will allow creation of 8080 source code from the surface of the tablet. The menu will consist of three parts:

1. A representation of the 8080 register architecture.

2. A list of general form instructions and psuedo op codes; mov, mvi, inx, org, end, etc.

3. A hexadecimal number chart to allow numeric entry.

This menu must allow each box to call a service subroutine. These routines are accessed by a table of three-byte jumps. The menu boxes will be decoded into a number that is a multiple of three. This multiple is then added to the start address of the table. The result is pushed into the program counter to vector the processor to the proper jump instruction. Boxes measuring .5x1 inch will result in 242 possible system instructions. These boxes may be assigned numbers by dividing the X displacement by 200. This will provide a column number of 0 thru 10. Only the integer is used (2200/200 must be treated as equal to 10). The Y displacement is divided by 100. This will provide a row number from 0 to 21. (Note: 2200/100=21). To obtain number values for each box in increments of three, multiply the row number by 11, add the column number and multiply by 3. This result is now added to the origin of the jump table and pushed into the program counter. From this point the function of the box is defined by the service routine.

Summagraphics can supply 8080 source-code initialization, control and input routines for the parallel transfer of data to and from the Bit Pad. All that is required to use the system is one parallel input port and one parallel latching output port. An active imagination will keep you busy for many evenings with the Bit Pad. ■

Menu Truth table

Y data

bit 8	bit 7	bit 6																
1	1	1	p	q	r	s	t	u	v	w	x	y	z					
1	1	0		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
1	0	1	p	Q	R	S	T	U	V	W	X	Y	Z					
1	0	0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0	1	1	0	1	2	3	4	5	6	7	8	9	:	;		=		?
0	1	0		!	"	.	$	%	&	'	(	)	*	+	,	-	.	/
0	0	1																
0	0	0																
bit 9			0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
bit 8			0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
bit 7			0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
bit 6			0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

X data

Ballad of Ole 360

or

Days and Nights in Frustration Center

Tune: Clementine

On the campus, in a building,
Almost treated as a shrine
Stood an IBM computer
That surely was no friend of mine.

Refrain:
Oh computer, Oh computer,
Oh computer, so serene,
I will never solve the mystery
Why to me you were so mean.

Every morning in apprehension
Took my place at pick-up time
Ole computer smiled and chuckled
Printed error on each line.

Refrain

Of teaching me the tricks of Fortran
Miss DeVore did despair
For every DO loop that I nested
I found I had a new gray hair.

Refrain

Said the Doctor, David Moursund,
Here's your bill for paper, sir,
For the extra cards that you punched
We had to buy a Douglas fir.

Refrain

In desperation punched my program
On a Playboy centerfold
Old compiler took a long look, gulped,
and knocked itself out cold.

Refrain

In jubilation said goodbye to
That machine that I did cuss
From now on I'll do my figuring
On a base-five abacus.

Robert Burghardt

The Real-World Connection: AC and DC Power Controllers

Neil Dvorak

The naked computer needs help relating to our real electrical world. Its processor communicates to other devices in parallel binary, which may be 4, 8, 12, 16, or even 32 lines, depending on the computer itself. However, man's instruments, extending the senses to measure the real world, usually present information in analog form, and sometimes this data is only a few millivolts (thermocouples for example). At the other extreme, some functions require the control of many amperes of current, and the power managed in these loads always dwarfs the microwatt signals active inside the typical microprocessor.

History

For years, industrial-grade computers have been interfaced with complex process-control systems. Sometimes the individual functions were backed up with electronic hardware to avoid disaster in the event of computer failure. In general, the entire technology is defined by the end-process or goal, and most systems tended to be custom-made. As a result, the art of reliable computer interfacing and operation was highly technical and known only by select circles of professional "system engineers" in the larger computer companies.

The proliferation of affordable small system computers has outspaced the availability of corresponding interface to the real world. However, well-packaged hardware systems now available to the hobbyist are specifically designed to enable the computer to actually control all those things it can manage so well.

Applications

Some processes do not require computing power, but others do, and a remarkable switchover to computer control is accelerating as microprocessor and interfacing costs come down. Many automotive engines will be under microprocessor control at the end of this decade. On a larger scale, commercial solar-energy installations require some computer brainpower for efficient usage. Or as an information assistant, a small computer with a video terminal can display dozens of variables simultaneously, eliminating the need or duplication of massive, expensive meter panels.

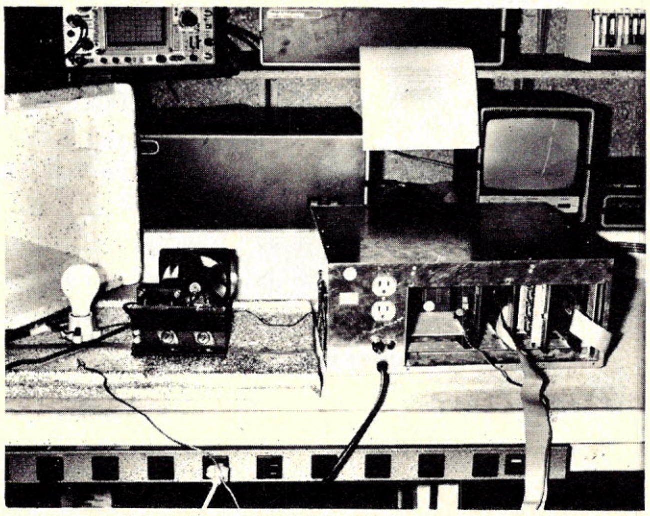
The several applications that follow detail some of the hardware items a user and programmer can expect to eventually encounter.

Climate Management In Buildings

Microprocessor-based systems are ideal for controlling environments with some degree of sophistication. Individual rooms can be adjusted with a duct damper in GFA (gas-forced air) units. Circulating hot-water systems are individually zone-controlled by a modulating value. Both operate from 24 or 115 VAC. Multiple remote temperature feedback is possible because most low-cost sensors can be conveniently used at any distance from the computer. Buildings designed to take advantage of solar energy can be programmed to circulate the warm air from one side over to cooler areas when conditions permit. Those who desire to use their computer only in a passive manner will note that it can be programmed for data-logging. Gas pressure is usually fairly well regulated, for example, and simple measurement of the main furnace gas solenoid valve gives a cumulative total of natural-gas consumption. For electric heating installations, line-voltage regulation is usually adequate to permit power measurement of any line branch. Use of a current probe provides an isolated DC voltage compatible for digitizing by an analog-to-digital converter. Thus electric power consumption can be tabulated. Devices such as heating or ventilation, when connected with a controller card, can be turned down should peak power exceed limits defined and charged by some power companies.

Positioning Systems

Special positioning machines are often controlled with stepping motors. These devices rotate a precise number of degrees upon change of a logic state. A DC Controller card and the appropriate software routine permit motors up to several amps per phase to be driven. Other positioning systems, often hydraulic as well as electric, require an analog voltage for the position reference. A Digital-to-Analog



This lab set-up uses the Real World Interface and a temperature probe to control how long the lightbulb stays on, so as to cycle the temperature up and down at a predetermined rate. The printer output is at top.

Neil Dvorak, Diego, Inc., 1031 West Center, Denver, CO 80223.

Converter card can be programmed to deliver this reference signal. Such flexibility permits the simulation of conditions necessary for realistic testing.

Electronic Art

Computing power raises the art of light display to levels limited only by imagination. Using the 32-channel low-power I/O card, LEDs or bright small lamps may be orchestrated with self-changing routines. The audio enthusiast can monitor musical signals via the A/D card and branch off into control routines depending on frequency and level. A multitude of special effects, written in small assembly-language routines, can be managed with executive software written in BASIC. Square waves can be manipulated by the beginner and more complex waveforms by those who use the D/A output card.

Real-World Interface

Most industrial-grade interfaces come as a bin or card cage that accept various function modules or boards. All are common to a data-power bus which in turn is connected to a host computer I/O register, usually by a bus controller or buffer. Cards are randomly addressable and can be added or removed depending on the user's needs. Output functions result from a write operation by the computer to the appropriate card as defined by its address. Or the computer may read a parameter by addressing the appropriate input card.

The Digital Group RWI (Real World Interface) system was designed in similar fashion. The eight-bit bus structure allows easy connection to all microsystems in addition to a Digital Group I/O port. Boards are presently available for multichannel AC control, and DC control, A/D and D/A conversion, temperature measurement, and current measurement.

Output Devices

AC Controller. This board incorporates the equivalent of eight solid-state relays. These isolated switches can be used to control relatively heavy loads such as motors, lamps, AC solenoids, and heaters. On-board jumpers determine the address of the controller card, while each bit of the data word corresponds to one of the eight loads to be energized or not. Individual loads up to 12A RMS may be controlled.

DC Controller. Designed to drive lamps, relays, DC or stepping motors, or other medium-current DC devices from an external DC power source. Each switching transistor is non-isolated but latched, and addressing is the same as the AC Controller card. Each of the eight-output transistors can handle up to five amps.

Analog Output Card. This card permits the generation of analog data to many devices, such as oscilloscopes, amplifiers, motor drivers, XY recorders, and other voltage controlled devices. Unlike some multichannel converters which require cumbersome software refresh, this unit contains its own memory and refresh circuitry. Up to 16 channels of analog output are available, and output range is user-adjustable.

Input Devices

Analog Input Card. Contains an instrumentation amplifier, an input multiplexer that can select up to 16 single ended or 8 differential mode signals, a track and hold buffer, and a high speed A/D converter. Amplifier gain and input, bipolar or monopolar, are user selectable. Low input drift permits connection of transducers such as thermocouples and the short conversion time allows digitizing most of the entire audio spectrum.

8-Channel Temperature Card. Can readily measure most environments from -40° to 100° C. Adjustments in the conditioning circuitry permit the user to select slope and zero point. The transducer itself may be located at any length from its conditioning circuitry. Installation requires a matching number of input channels on the analog to digital converter card.

The 32 I/O Card. Was developed to handle non-TTL-compatible logic levels and loads. Depending on instructions from the computer, the board will operate in several modes: high-level input sense, medium-current switch to ground, or a combination of the first two modes where the sense circuitry is enabled to test the loads for open-circuit burnout. Thirty-two multi-purpose channels are provided, all uniquely addressable. As an input device, up to 30 volts may be sensed. As an output device, up to 30 volts at 40ma may be switched per channel. In the load-testing mode, any of the outputs can be checked by disabling the drivers and testing for open circuit load voltage.

Current Probe Card. Enables the user to measure AC currents without making electrical contact to the line being measured. It is extremely useful in monitoring power consumed by appliances, motors, air conditioners, and even major branches in home wiring if the sensor is placed at the fusebox. This would find application in such systems as reducing costs when using a peak-demand power meter. An internal preamplifier outputs a filtered DC voltage proportional to average current. The conditioning circuitry comes as a plug-in card which allows three additional current probes to be added simply by the purchase of an upgrade kit.

Demonstration Program Environmental Test Chamber

```
1   INPUT L, U      L=LOWER LIMIT; U=UPPER LIMIT TEMPERATURE
5   OUT 5,4         RWI DATA BUS IS CONNECTED TO PORT 5
10  OUT 4,255       ADDRESS INFORMATION SENT ON PORT 4
15  OUT 4,64        TURN ON HEATER; ITS ADDRESS IS 64
20  GOSUB 91        INPUT THE TEMPERATURE
25  IF T  U THEN 30 ELSE 20
30  OUT 5,0
35  OUT 4,255       TURN OFF THE HEATER
40  OUT 4,64
45  GOSUB 91
50  IF T  THEN 5 ELSE 45
91  OUT 5,0         SELECT TEMPERATURE SENSOR ON CHØ
92  OUT 4,255
93  OUT 4,254       JUMPERED ADDRESS OF A/D CARD
94  V=INP (5)       PUT DATA FROM INPUT PORT 5 INTO LOCATION V
95  T=V/2.55        CONVERT TO DEGREES C° (5Volts = 255DEC = 100° C)
96  PRINT T
97  RETURN
```

Using the Digital Group RWI System

The author decided to construct a small test chamber to demonstrate the control functions of such an interface. Heat cycling is an almost indispensable tool for quality control and the programmable feature allows flexibility for testing a variety of components and assemblies.

Major system components for such a task are a temperature conditioning card, an analog-to-digital converter card, an AC controller card, and finally a CPU interface card. The latter would be common to any configuration used. Most of the boards are preliminary versions but documentation was adequate to verify system calibration using the boiling and freezing point of water and the "Input Temperature" subroutine in the software table.

A styrofoam picnic cooler was chosen as the chamber, using a 100-watt lightbulb as the heat source. A 4-inch muffin pan was used to continuously circulate the air internally to eliminate temperature gradients.

After the hardware was configured the BASIC software was a snap. Most of the project time involved defining input and output locations and making the appropriate connections. The following program works either as a temperature cycling oscillator or classical set-point thermostat. Loads less responsive than the small chamber described would require heat anticipating routines and other software techniques of professional process control for optimum utilization.

Components

The components described here are available from The Digital Group, Box 6528, Denver, CO 80206. They range in price from $38 kit ($45 assembled) for the 4-channel Current Probe Card with one probe, to $125 kit ($150 assembled) for the 8-channel AC Controller. A cabinet, with motherboard, power supply and paralled CPU interface, is $199.50 kit ($260 assembled). ■

Interfacing with the Outside World

Farhad Forbes
Harry Garland
Takao Tsuiki

The Telltale Heart: EKG Processing by Microcomputer

Analog signals that once were processed only by linear circuits are falling increasingly prey to digital processing by microcomputer. The idea of digital signal processing is really quite simple. A continuously varying analog signal is rapidly sampled. Each sample is converted to a digital value by an analog-to-digital converter. These samples are then collected as digital words by a microcomputer system, processed according to some algorithm, and output as a new set of digital words. A digital-to-analog converter is then used to create the processed analog waveform from these output samples.

Due to the limits of processing speeds in the microprocessors available today, it is not practical to perform real-time digital processing of high-frequency or rapidly changing analog signals. Signals containing frequencies of less than a few kilohertz however, can be processed quite effectively. One such signal is the electrocardiogram (EKG) produced by the contractions of the human heart.

Farhad Forbes, Harry Garland, and Takao Tsuiki, Cromemco Inc., 2400 Charleston Rd., Mountain View, CA 94043.

Digital signal processing has several advantages over conventional analog signal processing, the most notable being reliability and insensitivity to temperature and varying device parameters. In the past, minicomputers were needed for this sort of signal processing, but because of their high cost it was usually uneconomical to dedicate an entire minicomputer facility to a single application and, as a result, the analog data to be processed would have to be recorded on tape and processed later. Hence, real-time processing could not be achieved. Since it is possible today to custom-build a small minicomputer system to perform a specific function relatively inexpensively, the instrument measuring the analog data can have the digital processor built into it.

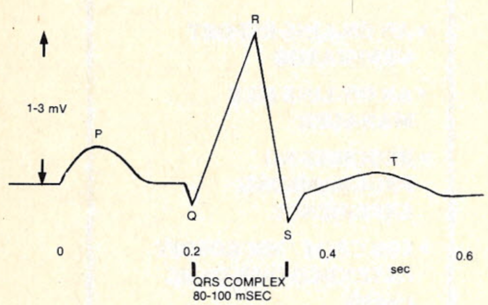


Fig. 1. Typical EKG waveform.

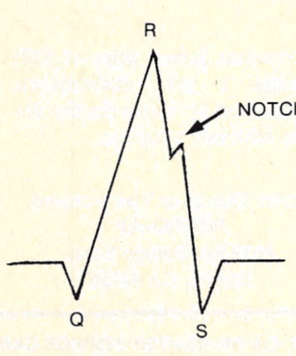


Fig. 2. The QRS complex of a high-frequency EKG showing a "notch."

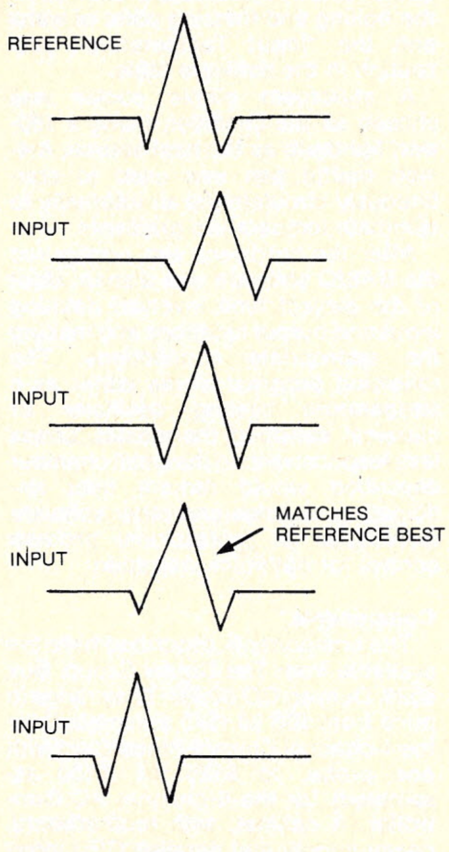


Fig. 3. The input QRS complex is shifted from right to left to determine the position in which it best matches the reference.

A Real-Time Averaging EKG Filter

Because electrocardiogram (EKG) signals are very low amplitude signals they usually contain fairly high levels of noise, especially at frequencies above 100 Hz. (See box for a more detailed explanation about the EKG and the various problems that affect its recording). Thus, in conventional EKG analysis the recorded signal is band-limited with a low-pass filter to about 100 Hz to remove the noise. The disadvantage of this technique, however, is that high-frequency components of the EKG signal are attenuated along with the noise, thus distorting the signal.

To filter out the noise and at the same time preserve the high-frequency components in the signal, an averaging digital filter was implemented using a Cromemco microcomputer and a Cromemco D+7A 8-bit A/D and D/A card. A block diagram of the system is shown in Figure 4. The program is stored in a 1K-byte 2708 EPROM. An additional 1K of RAM is used for the various data buffers used during program execution.

As is shown in the flow chart of Figure 5, there are four distinct steps in the data processing: (1) detection and input of the QRS complex of the EKG (2) alignment of the QRS complex before averaging (3) addition of the aligned input QRS complex to the sum of previously received QRS complexes (4) computation of the average and output of data.

The input phase of program execution is entered when a manual external switch generates an interrupt (see Figure 5). This informs the microprocessor that processing can begin and that the next N QRS complexes are to be averaged. In the program, N was nominally set to 16, but with minor modifications can be set to any power of 2.

Using a programmed input routine, the EKG data is sampled at the A/D

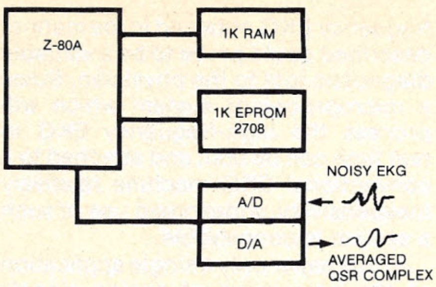


Fig. 4A. The Cromemco Z-1 system as used for EKG processing.

input port at a sampling rate of 1 kHz to 2.5 kHz. Since, irrespective of the sampling rate, only 256 samples are inputed, the sampling rate determines how much of the EKG waveform will be averaged. For example, a sampling rate of 1 kHz implies a sampling period of one millisecond and thus 256 msec of the EKG waveform are inputed. The maximum frequency response for a sampling rate of 1 kHz would be 500 Hz.

Also during the input phase, detection of the QRS complex takes place, so that at the end of the input phase, 256 samples of the QRS complex are stored as 8-bit words in the input buffer.

Next, the QRS complex just received needs to be aligned with a reference QRS complex before it can be averaged (see box for explanation). This is achieved by shifting the QRS complex in the input buffer relative to a reference QRS complex until the best match (correlation) is obtained. At this stage the input QRS complex is ready to be averaged.

The aligned 8-bit data in the input buffer is then converted to 16-bit two's complement data and is added to the corresponding data in the sum buffer where each sum is stored as a 16-bit word. The three steps mentioned above are then repeated N times until the sum buffer contains the sum of N QRS complexes.

The data in the sum buffer is then divided by N and loaded into the output buffer (step 4, Figure 5). The averaged QRS complex now stored in the output buffer is then output to the D/A port and can be monitored on an oscilloscope. The microprocessor repetitively outputs the output buffer until the external switch is depressed again, when a new average is computed and displayed.

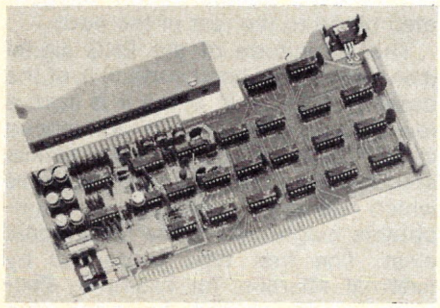


Fig. 4B. Cromemco D+7A analog interface board, used to interface EKG signals with the microcomputer.

The section of the averaging process which requires the longest processing time is where the input data is aligned and then summed (steps 3 and 4 in Figure 5). Processing time was minimized by using the Z-80 registers wherever possible instead of RAM. Also, by restricting the number of QRS complexes in the average to being a power of 2, so time consuming general purpose division routine is required.

Steps 1, 2, and 3 in Figure 5 take approximately 500 msec to process at a 4MHz clock rate. This is adequate for most EKG signals since the time interval between successive heart beats usually ranges from 800 to 1200 msec.

Results and Conclusion

An EKG waveform simulator was used along with a noise generator to simulate the noisy EKG's which were required to obtain the results described below. The noise-free output from the EKG simulator is shown in Figure 6.

An EKG with noise added in is shown in Figure 7A. Figure 7B shows the level of noise present in Figure 7A, while Figure 7C shows an average of 16 QRS complexes. Only the QRS complex is shown in the average. Figure 7D shows the same 16 complexes average on an expanded time-scale. Both these figures show that much of the noise is removed in the average.

In Figures 8 and 9, 32 and 64 QRS complex averages, respectively, are taken of the same noisy EKG shown in Figure 7A. Comparing Figures 7D, 8 and 9 it is seen that the 64 complex average in Figure 9 is the best of the three, thus showing that increasing the number of QRS complexes does improve the average.

Finally, Figure 10 demonstrates the necessity for correctly aligning the QRS complexes before averaging. Figure 10B shows a 16-complex average for the EKG in Figure 10A. No alignment was done and the data was averaged as soon as the QRS complex was detected. In Figure 10C, however, each QRS complex was aligned with a reference before averaging. Note that there is a considerable loss of detail in Figure 10B and that the higher-frequency components are much better preserved when the alignment was carried out as in Figure 10C.

These results show that averaging is an effective technique for filtering substantial levels of noise in the EKG. Moreover, using an alignment process to line up the input QRS complexes before averaging greatly improves the average, and helps preserve the high frequency components.

Because high-frequency notches in the EKG are known to be an early indicator of heart disease, a high-

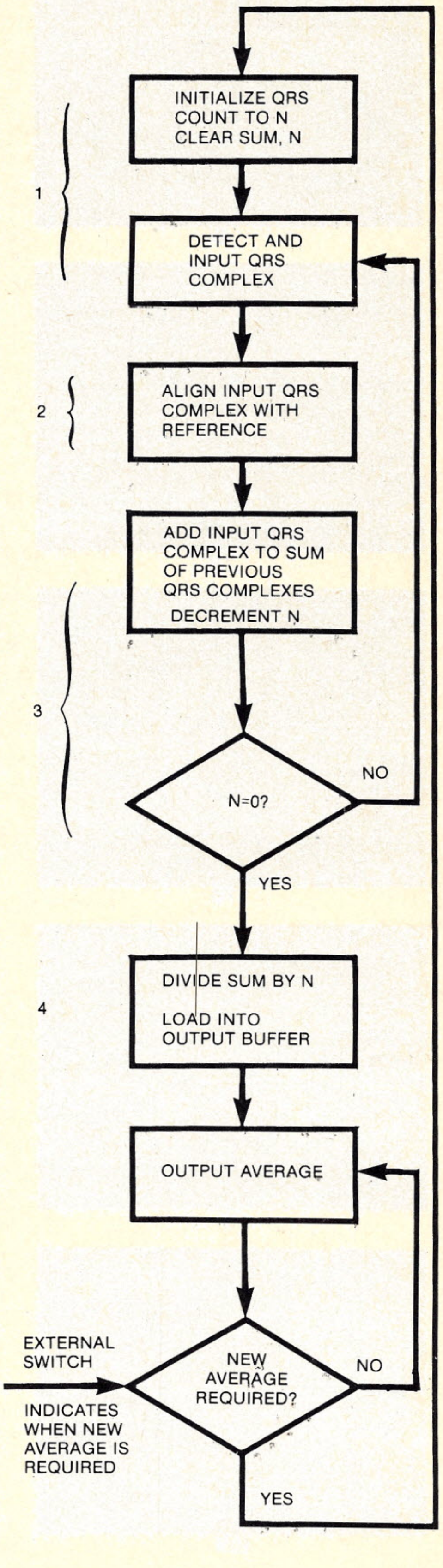
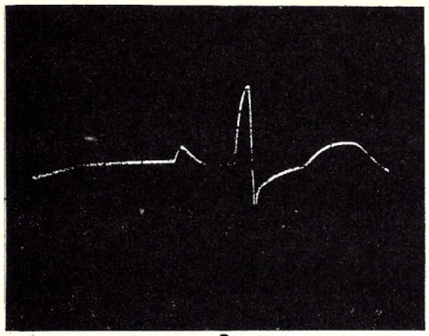
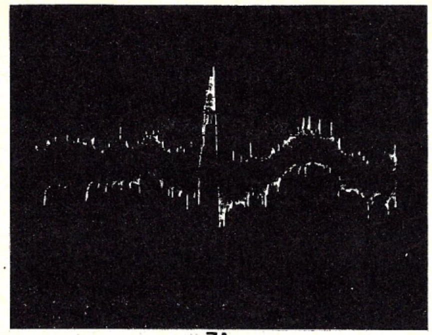


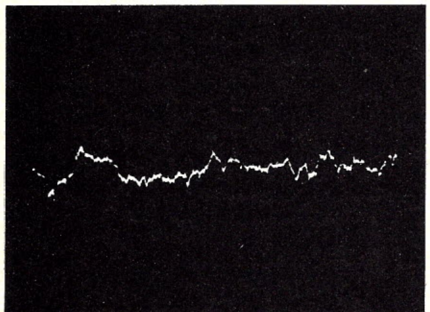
Fig. 5. Flowchart showing the four steps in processing the EKG data.



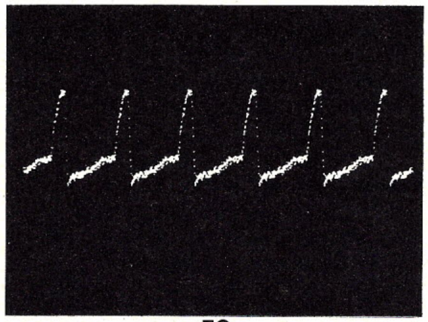
6



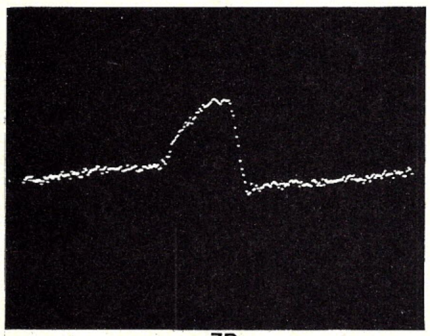
7A



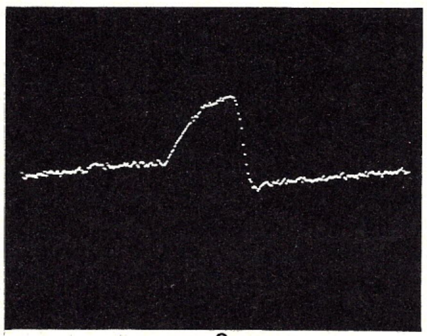
7B



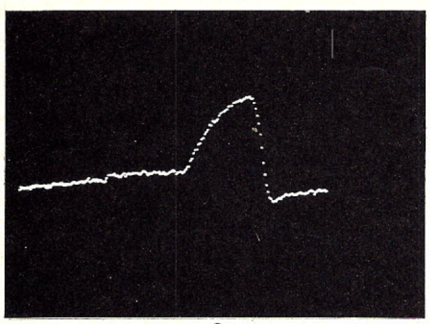
7C



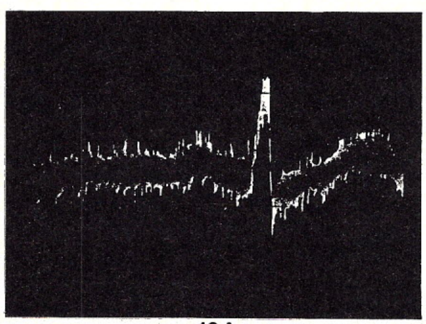
7D



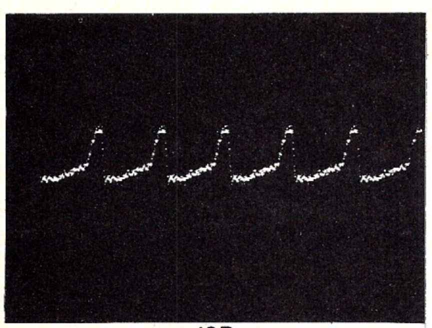
8



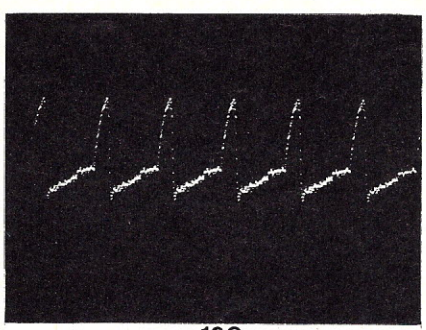
9



10A



10B



10C

Fig. 6 shows the noise-free output from the EKG simulator; Fig. 7A, an EKG with added noise; Fig. 7B, the noise level present in Fig. 7A; Fig. 7C shows an average of 16 QRS complexes; Fig. 7D, same as 7C, on an expanded time-scale. Figs. 8 and 9 show averages of 32 and 64 QRS complexes. Fig. 10B shows an average of 16 QRS complexes, without alignment for the EKG in Fig. 10A; Fig. 10C shows the result of alignment.

frequency EKG obtained in the manner described could prove to be a valuable diagnostic tool to the physician. Since a microprocessor system which will process the high-frequency EKG in real time can be built and attached to a conventional EKG machine relatively inexpensively, widespread use of such a system is conceivable.

In this paper only a single application of the processing of analog signals with a microcomputer was discussed. Microcomputers have and will continue to be used in a variety of signal processing applications. In fact, as they get less expensive, more sophisticated, and faster, complex digital signal processing that once could only be handled by high priced mini-computers will be performed in real time by small special purpose microcomputer systems at a fraction of the cost.

References:

1. This work was carried out at Stanford University under the support of Hitachi, Ltd.
2. H. V. Pipberger et al, "Recommendations for Standardization of Leads and of Specifications for Instruments in Electrocardiography and Vectorcardiography". *Circulation,* Vol 52, No. 8, p. 11, August 75.
3. P.H. Langorer et al, "Wide-Band Recording of the Electrocardiogram and Coronary Heart Disease," *Am Heart I,* Vol 86, P. 308, 1978.
4. N. Thompson, personal communication.
5. L. Jansson et al, "Noise Reduction of EKG by Averaging." *Scand J Clin Lab Invest,* Vol 35, p.p. 781-787. 1975.

The EKG

The electrocardiogram (EKG) is a record of the potential differences caused by the conduction of electrical impulses within the heart. These potential differences are most conveniently measured through electrodes placed at various points on the surface of the body. A sketch of a typical EKG is shown in Figure 1. Although each of the three segments — the P wave, the QRS complex and T wave — are useful for diagnosing heart ailments, the QRS complex indicates the maximum activity of the heart and is thus, according to cardiologists, a particularly useful segment to analyze. (During the QRS complex a vital physiological function is performed — the ventricles of the heart contract, and the left ventricle pumps freshly oxygenated blood to the rest of the body.)

The magnitude of the EKG signal measured at the body surface is of the order of 1 to 3 mV. Since it is such a low-voltage signal, it is influenced quite significantly by extraneous noise sources. Two types of noise sources which generate voltages of comparable magnitude to the EKG signal exist. The first type is caused by external sources; for example, electrode noise caused by poor skin contact. The second type is caused by the electrical signals produced by

muscle contractions within the body.

Because the major part of the EKG spectrum is below 40 Hz and because much of the noise is of relatively higher frequency, the bandwidth for conventional EKG analysis has been set at 100 Hz by the American Heart Association.[2] However, some researchers, such as P.H. Langner[3], N. C. Plowers[4], and N. Thompson[5], have proposed that there may be considerable diagnostic value gained by extending the frequency response of the EKG. They have suggested that, by increasing the frequency response to 500 Hz, or 1 kHz, high-frequency notches in the QRS complex as shown in Figure 2 can be observed in patients with heart disease. These notches are not visible in conventional EKG recordings during early stages of disease, and thus high-frequency EKG records could potentially provide an early indication of conduction problems in the heart.

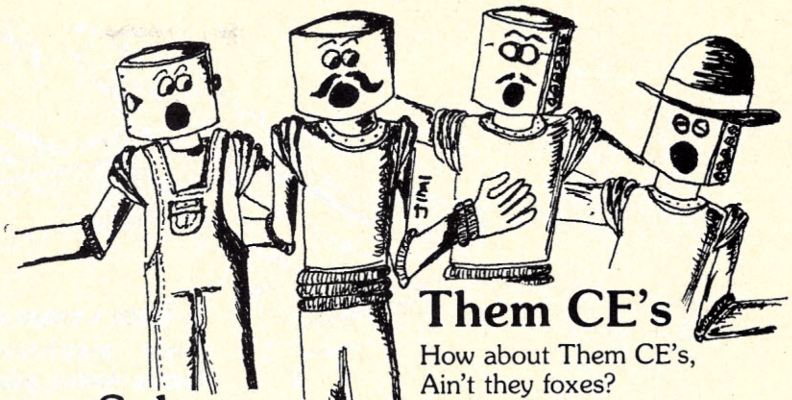
Hence, it is necessary to use a technique that would remove the noise, but which would preserve the higher frequency components in the EKG. A method suggested by L. Jansson[6], of the University of Lund, Sweden, is to digitize the analog EKG signal and average a predetermined number of successive QRS complexes. This technique takes advantage of the fact that even though there may be some coherency in muscle noise, most of the noise in the EKG is uncorrelated and will average out to zero if a sufficient number of QRS complexes is used in the average.

In the high-frequency EKG, because the main region of interest is the QRS complex (the duration of which is 80-100 msec) only this portion of the EKG need be analyzed. Thus, some method is necessary through which the QRS complex will be detected and aligned with the other QRS complexes before the average is taken. The alignment process is quite critical since any misalignment causes loss of detail in the averaged EKG and hence any advantage gained in using averaging to detect high-frequency components in the QRS complex will be lost.

Dr. Noel Thompson[5] has suggested that each incoming QRS complex be aligned with respect to a reference before being averaged. The principle is shown in Figure 3. The new input QRS complex is shifted with respect to a reference complex and for each shift a matching error is computed. When a minimum error is obtained it indicates that in this position the input QRS complex most closely matches the reference QRS complex and can then be included in the average. In the present work a microcomputer is used to perform this shifting operation, measure matching error, and compute the average EKG waveform. ■

Them ADP

William J. Wilton



Them Salesmen

How about Them Salesmen,
Ain't they deals?
Doin' they softshoe
In front of them wheels.

Floppin' them flipcharts,
Tellin' them lies,
Wowin' them management
With pie in the skies.

Look at Them Salesmen
Doin' they spiels,
Flashin' them credit cards
And wheelin' them deals.

Grabbin' them dinner checks,
Tellin' them jokes,
Fobbin' them Kludges
Off on innocent folks.

How to be a Salesman
And risk perdition?
Just let your sins
Be sins of commission.

Them Peripherals

How about Them Card-floggers,
Ain't they a laugh?
Suckin' up holey-cards
And spittin' out chaff.

Humpin' them Hollerith,
Doggin' them decks,
Sockin' funny punches
Into payroll checks.

Look at Them Printers,
Hotter'n Satan,
Bangin' they reports
On they platen.

Them online Clunkers,
Ain't they duffers?
Got them mainframe
Full of buffers.

How to be a Paper-raper,
Quicker'n sin?
Just grabba peripheral
And plug it in.

Them CE's

How about Them CE's,
Ain't they foxes?
Got they heads
Inside them boxes.

Makin' flakey fixes,
Hookin' up plugs,
A-gittin' them glitches,
A-gittin' them bugs.

Look at Them CE's,
Ain't they a case?
Got extra pins
In they interface.

Them wire wizard's wonder
Never ceases,
Got them mainframe
Tore to pieces.

How to be a CE?
Go on and do it.
Grabba hunka hardware
And sock it to it.

Them Coders

How about Them Coders,
Ain't they grand?
Got they templates
In they hand.

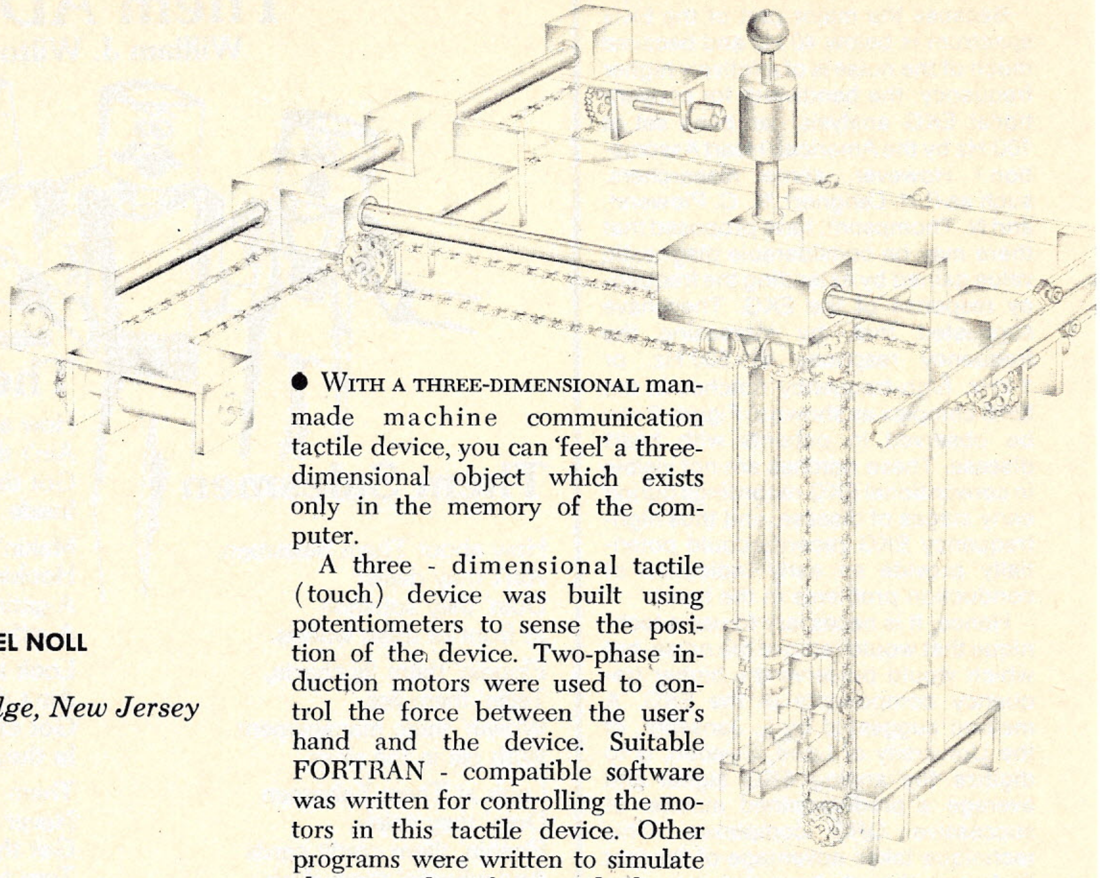
Them deadline doggers
Bustin' they humps,
Slashin' them O's
And doggin' them dumps.

Look at Them Hex-happies,
Ain't they guns?
Doin' it all
With zeroes and ones.

Them pointy-head crew'll
Give you fidgets
Calculatin' *pi*
To eighty thousand digits.

How to be a Coder?
Sin worse than sodomy.
Go out and gitcha
Self a lobotomy.

Rendition of *Them ADP* can be considerably enhanced by a special down-home algorithm achieved by stompin' them feet while clappin' them hands (a co-routine executable by any general-purpose programmer).



MAN-MACHINE TACTILE COMMUNICATION

Man, using a simple tactile device, can feel and identify shapes and objects existing only in the memory of a computer. Possible applications are cited.

By A. MICHAEL NOLL
AT&T
Basking Ridge, New Jersey

● WITH A THREE-DIMENSIONAL man-made machine communication tactile device, you can 'feel' a three-dimensional object which exists only in the memory of the computer.

A three - dimensional tactile (touch) device was built using potentiometers to sense the position of the device. Two-phase induction motors were used to control the force between the user's hand and the device. Suitable FORTRAN - compatible software was written for controlling the motors in this tactile device. Other programs were written to simulate objects and surfaces and also to position the tactile device at a specified point.

Results thus far suggest that tactile man-machine communication is useful for "depicting" surfaces and objects which would be virtually impossible to display visually. Man-machine tactile communication also has potential as a practical scheme for computer "graphics" for the blind. In addition, the non-blind have here a possible scheme for a better and totally different "feel" for computer graphics.

Taken from the Dissertation submitted to the faculty of the Polytechnic Institute of Brooklyn in partial fullfillment of the requirements for the degree Doctor of Philosophy (Electrical Engineering — 1971).

Reprinted from the July/August 1972 issue of SID Journal

Tactile continued...

Introductory Remarks

ALTHOUGH ONE CAN JUSTIFY a negative impression towards using stereoscopy for most displays of scientific and technological data, stereoscopy sems to become more important in those fields which rely more heavily on graphical presentations, such as architecture and design.[1] However, even stereoscopic presentations sometimes do not seem sufficient for many man-machine communication applications in these fields. As an example, the designer needs a computer-controlled "something" to help him mold shapes or forms using his hands and the sensation of touch. Thus, the temptation grows to explore the potential of new sensory modalities as new communication channels between man and machine in applications where graphical communication would not be sufficient or appropriate. Perhaps the feeling that computer graphics has been given too much emphasis in its role as a form of man-machine communication is justified. The blind, for example, have learned how to develop and exploit all sorts of non-visual communication abilities so that they can live most effectively in an otherwise visually-oriented world.

The above-stated possible needs of the designer for tactile communication coupled with the experience gained in investigating and designing a three-dimensional input device for use in man-machine communication indicate that the tactile communication channel would perhaps be suitable as a new form of man-machine communication.[2] The three-dimensional input device helped the user specify the location of a point in a three-dimensional space in cartesian coordinates. If this device could be controlled by the computer so its resistance to motion could be varied, then the user would, in effect, be able to probe, by feel, the contents of a three-dimensional space. This probe would be only a single point and would be akin to poking around with a stick. It would hopefully be a significant test of the possible usefulness of a new man-machine communication channel. This tactile device could be used to augment the stereoscopic display for such tasks as latching on to a line or object in three dimensions. It could also be used in psychological investigations of interactions between the human tactile and visual communication channels. A tactile communication facility opens the door to a totally new man-machine communication channel.

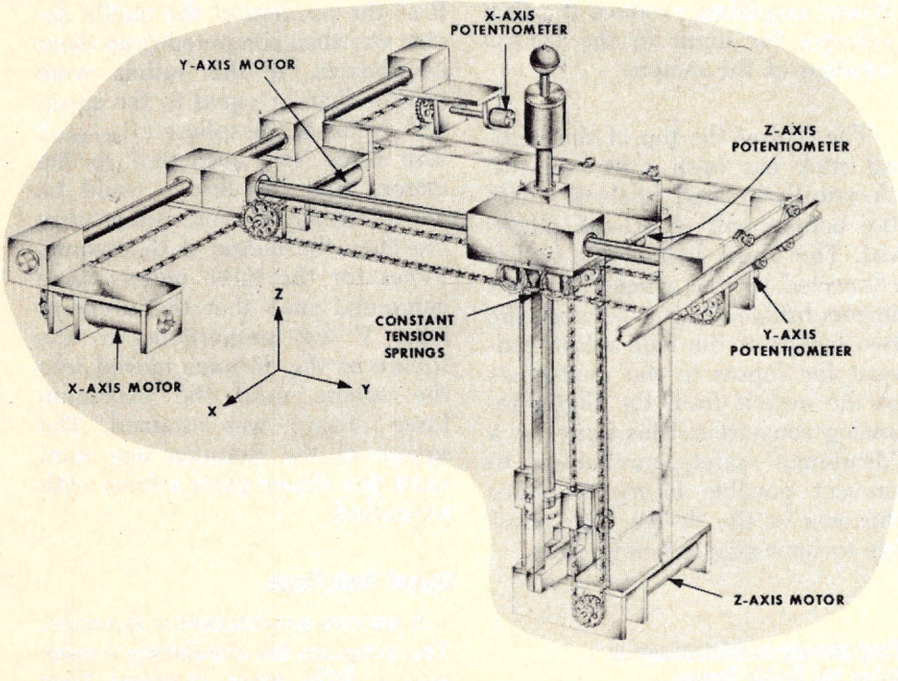


Fig. 1. Sketch of tactile device. The ball at the top of the vertical shaft can be moved within a 10-inch cubical space. The position of the ball is sensed by potentiometers while the force required to move the ball is controlled by motors.

Design of A Three-Dimensional Tactile Device

A COMPUTER HELPING AN individual feel some object which existed only in the memory of the computer could justifiably seem to be a "far-out" idea. One might imagine a computer-controlled, three-dimension, electromagnetic field with a hand-held ball suspended in the field as one possible implementation. But this is too esoteric. A down-to-earth hardware design is required to realistically evaluate man-machine tactile communication. Since a three-dimensional input device had already been designed and constructed, "simply" controlling the device so that the computer could vary the feel of the device, or even lock it in certain positions, seemed to be the best approach to the design of a tactile device.

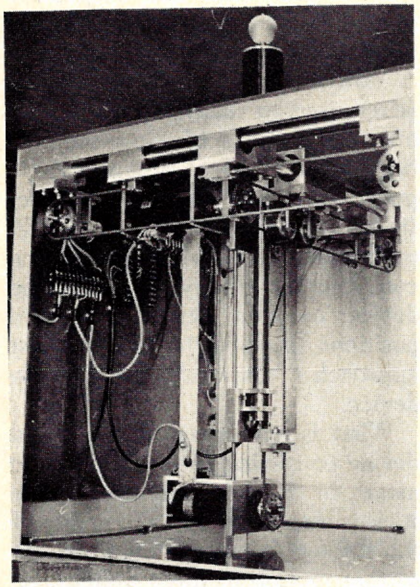
What is envisioned thus far is a device consisting of a stick, free to move in three dimensions. The stick is constructed in the image of the three-dimensional input device so that motion in three dimensions has been mechanically separated. This device is shown in Fig. 1. Chains and sprocket drives of potentiometers would be used to sense the position of the stick-like portion of the device held by the user. The device might be requested to resist motion for those applications in which the user is bumping into the surface of an object. In other applications, the device might be requested to assist motion to overcome its own inertia and friction so as to move as freely as possible.

Clearly the source of force control of the device would therefore have to be able both to resist motion and to assist motion. A motor with an electrically-reversible direction of rotation meets these requirements. Three such motors connected to their own sprockets would supply the assistance or resistance to motion of the device. A linear force of about twelve pounds would be the required maximum

force to simulate bumping into a fairly rigid object. Linear bearings would be used to minimize friction. More details about the final design of this tactile device form the remainder of the material in this section.

The mechanical design requirement was imposed that the vertical shaft when fully extended would not deflect more than 0.015 inches in any direction under a maximum force of 12 pounds. A defection of 0.01 inches was determined experimentally to be just noticeable to the human hand so that this deflection requirement was most reasonable in these subjective terms.

Equations for the maximum deflection of supported beams and cantilevers were used to calculate the theoretical deflections for the device when fully extended.[3-4] This theoretical analysis indicated that the shafts forming the major structural members of the device would have to be about 1 inch in diameter to meet the maximum-deflection requirement. A photograph of the interior of the device is shown in Fig. 2.



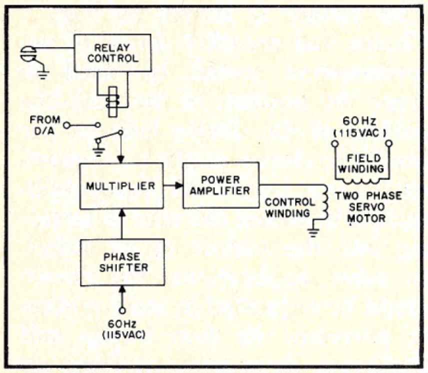


Fig. 2. Photograph of three-dimensional tactile device.

Fig. 3. Block diagram of motor control electronics. Separate phase shifters, multipliers, and power amplifiers are used for each of the three motors. The ball at the top of the tactile device has been split in half so that the user's fingers bridge the gap between the two halves and cause the relay control to operate. Thus, the user must be holding the ball for the motors to be energized.

Two-phase 60-Hz 10-watt induction motors were used to supply the forces needed to control the tactile device. The direct-current outputs from three digital-to-analog converters used to control the motors are converted to 60-Hz voltages at phases of either +90 degrees or −90 degrees depending upon the desired directions of rotation. This dc to ac conversion is accomplished by multiplying the dc voltages by 60-Hz ac voltages which have been shifted 90 degrees relative to the field winding voltages, as shown in Fig. 3. The signs of the dc voltages are retained in the multiplications so that the directions of rotation of the motors can be controlled also. Power amplifiers produce the final voltages for input to the control winding of the motors.

The ball at the top of the vertical stick has been split into two electrically-isolated halves, with the bottom half at ground potential. The upper half of the ball is connected so that when the user's fingers bridge the gap between the two halves of the ball, relays connect the inputs to the multipliers to the output from the digital-to-analog converters. This serves as a "dead-man" safety mechanism to prevent possible injury to either the user or the device, due possibly to some programming error.

Programming and Experience in Using the Tactile Device

THE SOFTWARE REQUIRED for the tactile device must simply input

Tactile continued...

the position of the device and output signals to control the motors. Since the position of the device is indicated by three potentiometers, a subroutine for inputting the value of a knob or potentiometer is called three times to input the values of the three potentiometers. The motor output is accomplished by a subroutine for simply outputting three numbers to the three digital-to-analog converters which control the three motors. Thus, the guiding philosophy of FORTRAN-callable subroutines for use in FORTRAN programs for real-time interactive man-machine tactile communication was preserved, and all the program described below were written in FORTRAN using these two subroutines for communication to and from the tactile device.

One of the simplest and perhaps most basic shapes is the sphere. The tactile device was programmed to simulate a rubbery sphere suspended in space. The three-dimensional coordinates, X, Y, and Z, of the position of the tactile device were inputted to the computer which then expressed these coordinates relative to the center of the sphere. The radius R of the position of the tactile device was then computed from these coordinates. If this radius were greater than or equal to the specified radius of the sphere (R_{SPHERE}) zero force was outputted to the motors, and the device could be moved freely. If this radius were less than the radius of the sphere, forces for the three motors were computed such that the resultant force F was proportional to the square of the distance moved into the sphere until the maximum force (F_{MAX}) was attained. The square of the distance was used since this choice gives a force with a nice feel.

Second Basic Shape

A SECOND BASIC SHAPE is the cube. The program for simulating a cube was a little more involved than that for the sphere program because to compute the force the program had to know along which

axis the cube was approached. The cube was suspended in the three-dimensional space such that its faces were parallel to the three axis of movement of the tactile device. Thus, it was necessary to output a non-zero force to only one of the motors to simulate bumping into a face of the cube, while zero force was outputted to the other two motors.

The cube was simulated by first inputting the three-dimensional coordinates of the tactile device. These coordinates were then expressed relative to the center of the cube. If the tactile device were outside the cube, zero force was outputted to the three motors, and the computer determined along which axis the cube was being approached. As soon as a face of the cube was entered, a force proportional to the square of the distance moved into the face was calculated until the maximum force was attained. The width of this square-law force region was variable so that the sponginess of the cube could be varied.

Users Felt Their Way Around

THE TACTILE DEVICE was programmed to simulate an object with a cubical exterior and a spherical interior. The cube and the sphere algorithms were used for these shapes. A cylindrical hole at the top of the cube allowed the user to enter the spherical interior. However, once the interior was entered through this hole, the hole was closed, and the user had to exit through a cylindrical hole in the side of the sphere. However, once the interior was exited through this hole, the hole was closed, and the user could re-enter the interior only through the hole at the top of the cube.

The cube program was modified to present a stereoscopic display of the edges of the cube and a dot representing the position of the tactile device. This was done to disprove the hypothesis that the user's "feel" of the cube would be strengthened if the user could simultaneously "see" both the cube and the position of the tactile device. Most users looked at the stereoscopic display as they "felt" the cube. But, after a short time they abandoned the display and simply felt their way around the cube by feeling the edges, falling off the edges, and sliding along the faces. Thus one quickly concludes that stereoscopic display is not necessary as an adjunct to man-machine tactile communication.

Identifying the Shapes

USERS WERE ASKED TO IDENTIFY the sphere or cube by feel alone and without being told what objects were available in the repertoire. Most users had difficulties in correctly identifying the spongy sphere although they quickly identified the cube. The major source of difficulty with the sphere was that the users nearly always slid off the surface since the sphere had a convex surface when felt from the outside. This difficulty did not occur with the sphere-within-a-cube since the inside of the sphere was a concave surface. Most users were able to explore the sphere-within-a-cube and correctly identify it along with the one-way cylindrical spaces joining the outside and inside.

Residual magnetism in the motors and leakage in the multipliers produced rotational resistance in the motors. This rotational resistance was increased by the gear train and, together with bearing friction, produced enough friction that nearly a half pound of force was required to move the tactile device. This was most bothersome to the users. Friction manifests itself in the differential equation governing motion of the device as a term proportional to velocity in a direction opposing motion. To overcome this friction, the first differences of the position of the device were computed and used as an approximation to the three-dimensional velocity of the device. These first differences were then multiplied by suitable experimentally determined constants, and the results were outputted to the motors in directions to assist movement of the device. These consants were the same for all three axes which was expected since the friction theoretically should be independent of direction. This velocity-dependent movement assist greatly increased the ease with which the device could be moved about.

Tactile continued...

Control of the Tactile Device

IN SOME APPLICATIONS, the tactile device might be required to remain at a specified fixed location in one or more of the three dimensions. If the user attempts to move the device, a restoring force must be applied to attempt to return the device to the desired position. The determination of this restoration force is a control problem. Although considerable information is available concerning the optimal control of some device, most of this information is theoretical and avoids practical problems.[5-6] Hence, a common-sense control law, combining optimal bang-bang control and linear control, was used as described below.

Control Law

If u represents the error in position of the device and $\dot{u}$ the velocity of the device to the origin of the $(u, \dot{u})$ plane. When approaching the origin, the device has an energy $c_1\dot{u}^2 + c_2u^2$. If a linear damped control law is used in a region near the origin such that $c_1\dot{u}^2 + c_2u^2 < E_{max}$, then the energy must decrease until finally the device stabilizes at the origin. Thus, a control law was programmed which applies optimal bang-bang switching if the state $(u, \dot{u})$ of the device is outside an elliptical region centered about the origin. If the state is inside this elliptical region, the motor control is $f = -k_1u - k_2\dot{u}$. With this procedure k_1 could be made large to give a large restoration force while the energy constraint could be chosen to insure that $|f| \leq N$ so that saturation would not occur. This control procedure was programmed with the time delay correction, but the best performance was achieved by removing the delay correction. The computer plotted the state space of the device defined by u and $\dot{u}$ so c_1 and c_2 could be easily determined by varying two knobs to produce a good trajectory to the origin. If the user moves the device, his hand feels a linear restoring force. No chatter or oscillation is present.

Tactile continued...

Future Possible Applications

THE WORK THUS FAR completed indicates that man using a conceptually simple tactile device can feel and identify shapes and objects existing only in the memory of a computer. Furthermore, the tactile device can be positioned by the computer to remain at a prescribed point. This demonstrates that the computer can be programmed to restrain the tactile device so that it can be freely moved by man over only a prescribed three-dimensional path or surface. These might seem to be meager results for extrapolating all sorts of possible future applications for tactile communication, in addition to predicting vastly more elaborate tactile communication devices. However, past experience strongly implies that science and technology have a startling ability to develop whole new fields with such unbelievable speed and in such completely unexpected ways that even the wildest extrapolations and predictions based upon present results usually seem conservative in a few years.

'Like Blind Man'

The tactile device presently constructed is concerned with computer control of the force felt at only one point within a three-dimensional space. This situation is similar to a blind person exploring and poking around three-dimensional shapes and objects with the tip of a hand-held pencil. It is most tempting to drop the pencil and grasp the object or feel the shape with one's complete hand and the tips of five human fingers. This would be possible with a computer-controlled tactile device which consisted of individual force control mechanisms for each finger and electronic or mechanical "things" for each finger tip in addition to mechanisms for controlling the overall motion of the complete hand. With such a future tactile device man could grasp objects and feel the surface texture of objects which existed only as equations or arrays of numbers in the memory of the computer.

The present and future devices for obtaining tactile communication from the computer could be augmented using a three-dimensional helmet-type display similar to that presently being used for computer-generated displays.[7] A mechanical linkage attached to the helmet senses the position of the helmet, and if the position has changed the computer recalculates the steroscopic display on the face of the two tubes. The display on the two tubes is seen by the user through half-silvered mirrors so that the external environment is also visible. In this way, the user might conceivably place his hand in the mechanism that is used for the tactile communication with the computer, and at the same time see both his hand and the computer-generated display. As an example, the user might see a computer-generated three-dimensional cube superimposed on a physical table. He could then move his hand towards the cube, feel the cube, grasp the cube through the force feedback from the tactile device, and even lift the cube from the table and feel its weight.

But what practical uses would there be for a system as elaborate as the preceding, or for that matter what possible uses would there even be for the simple man-machine tactile device described in this paper? One important use was mentioned before: namely, aiding and augmenting the "feel" of conventional man-machine communication through computer graphics so that when one latches onto an object in a display he also physically feels the latching on. Many psychological experiments come to mind that might investigate deliberately introduced clashes and offsets between the tactile and visual communication channels or the ability of a subject to identify objects by feel alone.

Hand-and-Fingers

A tactile communication device involving both the hand and the fingers would be an extremely useful design tool. With it one would be able to investigate the reaction of subjects to three-dimensional shapes and objects which could be simulated on the computer, for instance, a new design for a telephone handset. The manual dexterity of different individuals in performing assembly tasks could be scientifically investigated with computer-simulated objects, thereby resulting in an optimization of the design of an object from both an aesthetic and a functional viewpoint.

Aid to Handicapped

Perhaps the most humanistic use for a tactile communication device is as an aid for the handicapped in communicating with computers. A segment of humanity exists for whom the term "computer graphics" and all the comments about the desirability of man-machine graphical communication are completely meaningless — namely, the blind. With a tactile communication channel the blind would be able to feel the shape of graphs and other curves and surfaces and even objects. As a simple example, a blind person might hold the present tactile device while the device would be constrained by the computer so that it could be freely moved only along a prescribed three-dimensional surface or curve. If humans gifted with sight are able to identify shapes and objects by feel alone using the present tactile device, then the blind with their highly developed sense of touch and tactile memory abilities should perform significantly better.

Could 'Feel' Textiles

Perhaps the second most humanistic use for tactile communication is for communication from man to man and possibly, but not necessarily, involving computers as some form of intermediary. For this application, two humans located at two physically separate locations each with a tactile device would communicate with each other using the tactile devices and a communications network to link together the two devices. As a possible practical application a purchaser of cloth located in New York City could feel the texture of cloth produced by a textile manufacturer in Tokyo without physically transporting any cloth anywhere. A man-to-man tactile communica-

tion facility could certainly be augmented and coupled with facilities for the transmission of sound and images. Thus, the senses of vision, hearing, and touch would have been extended over great physical distances, and "teleportation" in one sense would be closer to reality.

Future Directions of Research

A two-dimensional tactile device has been constructed by experimenters at the University of North Carolina.[8] They used their device to demonstrate that force output from the computer can help students better "visualize" concepts in elementary electromagnetic fields. The tactile device described in this paper applies reasonably large forces in three dimensions and could be used to further study the usefulness of tactile communication as an educational tool. The usefulness of man-machine tactile communication as an aid to the blind must likewise be evaluated through carefully-controlled experiments using sighted subjects as a control group. Similarly, tactile communication must be evaluated for its usefulness in supplementing three-dimensional man-machine graphical communication. A unique opportunity exists here to evaluate the effectiveness of computer graphics for man-machine communication now that an alternative form of man-machine communication has been created using the tactile device.

Psychological intersensory conflict experiments introducing deliberate distortions of the visual field have been conducted in the past.[9] With the tactile device it would become possible to introduce independent distortions between the visual channel and the tactile channel. The tactile device using a three-dimensional force-measuring mechanism to determine the force exerted by the user's hand in moving the device might be used in investigations of motor skills involving hand movement. Thus, the tactile device could easily be the common tool in a host of new areas of investigations by perceptual and motor-skill psychologists.

In the hardware area, the design and implementation of a new tactile device embodying control of the five fingers through hydraulic mechanisms would allow the user to grasp and feel objects by program control. This ability would be most useful to designers. Such a device could be used to evaluate newly designed objects by simulating the physical feel and shape of the objects.

a. Further Thoughts on Tactile Communication

The graphical representation of complicated surfaces and solid objects has always been extremely difficult even using stereoscopic techniques. Grid lines could be drawn along the surface at regular intervals or dots could be scattered at random on the surface. Either way a considerable number of points would be required to represent adequately a surface with fine or complicated details, and large numbers of points create display problems in terms of flicker and interactive problems in terms of computation time. If one portion of the surface hides another portion then yet other problems arise in terms of the suitable graphical representation of the hidden surface. A stereoscopic display of the complete surface including the hidden portion is sometimes reasonably suitable, and the depth perceptive abilities of the viewer help him to separate in depth the different portions of the surface.

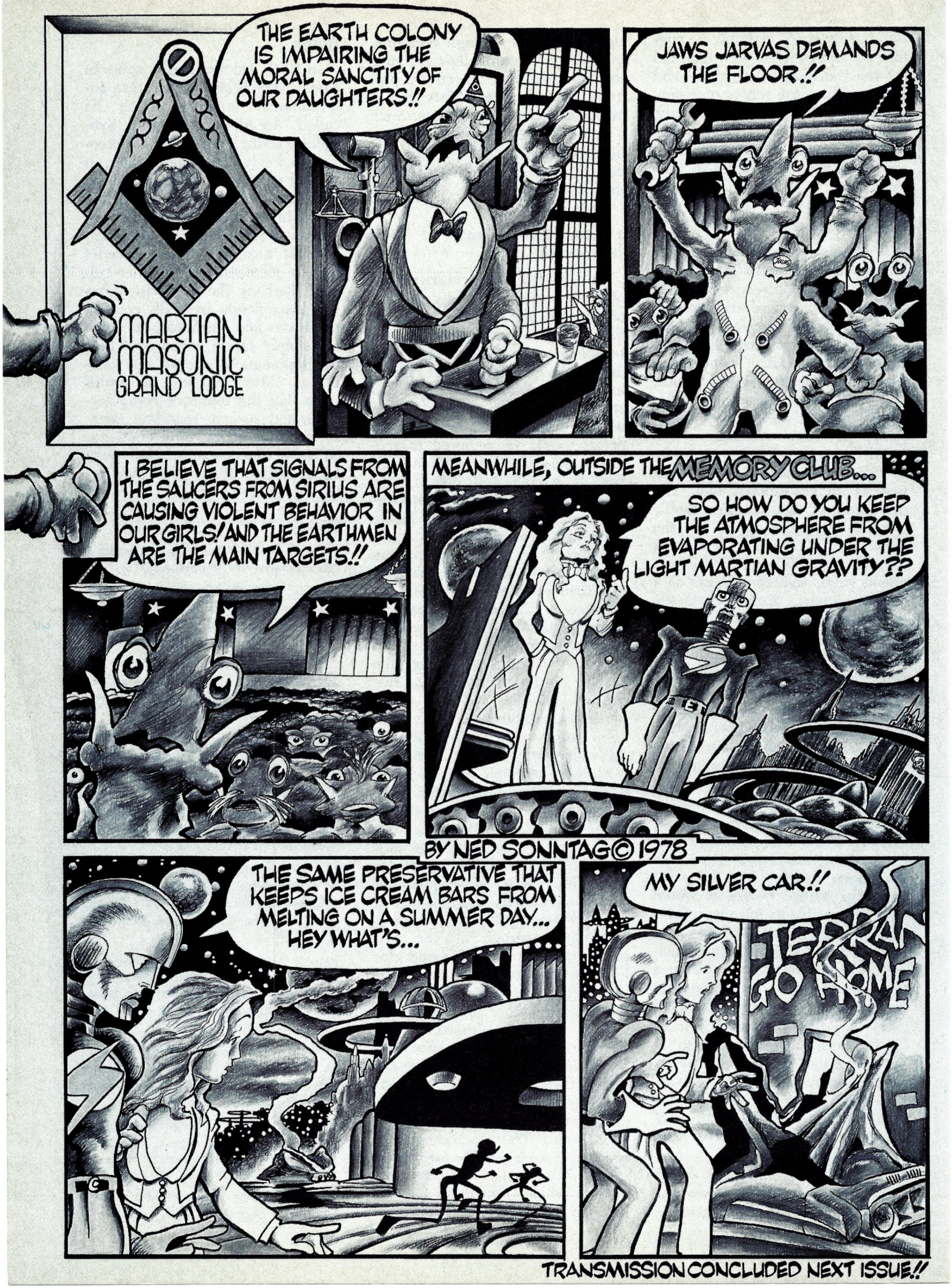
All these problems are completely circumvented when tactile communication is used to represent surfaces and objects, since the computer has to be concerned only with the position of the tactile device which is a single point in three-dimensional space. However, with a visual display the computer has to be concerned with the complete surface in all its fine and global details. With tactile communication the computer need only determine whether the tactile device is or is not "touching" the surface. The computer in effect can be concerned with all the fine details of the surface since it does not have to be concerned simultaneously with the global aspects of the surface also. Thus, tactile communication is most suitable for representing complicated surfaces and objects which would be far too detailed to represent graphically.

Epilogue

In 1932, Aldous Huxley wrote in *Brave New World* of a future entertainment medium which he called "An All-Super-Singing, Synthetic-Talking, Coloured, Stereoscopic Feely."[10] Perhaps there is indeed much more truth than most scientists and technologists would admit in the claim that today's science and technology are only acting out a script written decades ago by members of the other culture. Perhaps the best visionaries in science fiction are really creative scientists and technologists who are simply far in advance of new developments in science and technology. Most certainly it would seem that the topics of computer-generated speech, real-time interactive stereoscopy, and man-machine tactile communication were all predicted forty years ago by Huxley in his "feelies." ■

References

1. A. Michael Noll, "Real-Time Interactive Stereoscopy," (to be published).
2. A. Michael Noll, *Man-Machine Tactile Communication*, Ph.D. Dissertation, Polytechnic Institute of Brooklyn, June 1971 (available from University Microfilms).
3. Erik Oberg and Franklin D. Jones, *Machinery's Handbook*, Industrial Press Inc., New York (1970).
4. S. Timoshenko and Gleason H. MacCullough, *Elements of Strength of Materials*, D. Van Nostrand Company, Inc., Princeton, New Jersey (1949).
5. Irmgard Flügge-Lotz, *Discontinuous and Optimal Control*, McGraw-Hill Book Company, New York (1968).
6. L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*, John Wiley & Sons, Inc., New York (1964).
7. Ivan E. Sutherland, "A Head-Mounted Three Dimensional Display," *AFIPS Conference Proceedings*, Vol. 33, 1968 Fall Joint Computer Conference, pp. 757-764.
8. James J. Batter and Frederick P. Brooks, Jr., "GROPE-1: A Computer Display to the Sense of Feel," University of North Carolina, 1970 (to be published).
9. Charles S. Harris, "Perceptual Adaptation to Inverted, Reversed, and Displaced Vision," Psychological Review, Vol. 72, No. 6, 1965, pp 419-444.
10. Aldous Huxley, *Brave New World*, Doubleday, New York, 1932.


THE EARTH COLONY IS IMPAIRING THE MORAL SANCTITY OF OUR DAUGHTERS!!
MARTIAN MASONIC GRAND LODGE
JAWS JARVAS DEMANDS THE FLOOR.!!
I BELIEVE THAT SIGNALS FROM THE SAUCERS FROM SIRIUS ARE CAUSING VIOLENT BEHAVIOR IN OUR GIRLS! AND THE EARTHMEN ARE THE MAIN TARGETS!!
MEANWHILE, OUTSIDE THE MEMORY CLUB...
SO HOW DO YOU KEEP THE ATMOSPHERE FROM EVAPORATING UNDER THE LIGHT MARTIAN GRAVITY??
BY NED SONNTAG © 1978
THE SAME PRESERVATIVE THAT KEEPS ICE CREAM BARS FROM MELTING ON A SUMMER DAY... HEY WHAT'S...
MY SILVER CAR!!
TERRAN GO HOME
TRANSMISSION CONCLUDED NEXT ISSUE!!

INTERFACE AGE

The magazine that brings the computer home.

You want to learn all you can about personal computing. **Interface Age** tells it all. In simple, easy to understand language.

Each issue of **Interface Age** revolves around the latest information in the world of microcomputing. Articles, letters, new products, opinions, humor, experiments and new programs to help you choose and understand your computer and use it to full capacity.

Did you know your personal computer can –

- Help you start a business at home providing services to others.
- Increase your child's educational opportunity.
- Analyze real estate, stock market and other investments.
- Show you how to save money by computing and simplifying income tax & deduction records.
- Inventory your household items, store shopping lists and establish a recipe bank to plan daily meals.
- Control your small business with accounting and inventory control systems.

Technology has brought the computer to your world. **Interface Age** brings it home.

Order your subscription to **Interface Age** NOW with this convenient coupon. Save $7 over newsstand price.
12 Monthly Issues: $14 U.S.; $16 Canada/Mexico; $24 International. 24 Issues: $24 U.S.; $28 Canada/Mexico. 36 Issues: $36 U.S.; $42 Canada/Mexico.

Name ______________________
Address ______________________
City ____________ State ________ Zip ________
☐ Visa Card ☐ Master Charge
Acct. No. ______________ Exp. Date ________
Signature ______________________
Check or money order (U.S. Funds drawn on U.S. Bank)
Make checks payable to: INTERFACE AGE MAGAZINE
P.O. Box 1234 Dept. CC5 Cerritos, CA 90701
☐ *Please send information on issues available back to 1975.*

CIRCLE 222 ON READER SERVICE CARD

THEOREM-PROVING WITH EUCLID

Artificially Intelligent CAI in Extended BASIC

Thomas J. Kelanic

EUCLID is a tool for individualizing instruction in high-school geometry courses. With EUCLID, students can work at their own rates, and along their own routes, in constructing proofs. This article gives a brief description of the operation of EUCLID and indicates directions to be taken in its continuing development.

EUCLID is the fourth version of a theorem-proving computer program written by the author since 1972. The first three versions were written under the auspices of Project Solo and Solo-Works at the University of Pittsburgh. The present version was written at Taylor Allderdice High School, beginning in May, 1977. EUCLID consists of two computer program segments called EUCLID and GEOMTREE1, sharing three initially empty random-access files called PROFILE, CONFILE, and PARTRUE. Together, EUCLID and GEOMTREE1 contain 678 lines of Extended BASIC occupying 14K of memory. An additional 9K of memory is reserved for all three files.

Syntax

A "set" is a character string consisting of the first three letters of the corresponding term from geometry. SEG, ANG, and TRI are sets corresponding to segment, angle, and triangle, respectively. The symbol # denotes "the measure of," and is used to form the additional sets #SEG, #ANG, and #TRI, representing length, degrees, and area, respectively.

A "relation" is either CON for "is congruent to," or "=".

An "identifier" is a character string consisting of one, two, or three characters, located one blank space to the right of a set, and is used to identify a particular set. The triangle with vertices A, B, and C is identified as TRI ABC. All identifiers for segments and angles, but only the first triangle found in a string, are alphabetized by the program for unique identification. For example, SEG BA is changed to SEG AB, ANG BVA to ANG AVB, and TRI CBA TRI DEF is changed to TRI ABC TRI FED. Identifiers may contain any character except "&" and blank characters.

A "sentence" is a character string containing a set, identifier, relation, set, identifier sequence, with blank characters serving as delimiters. An example of a sentence is SEG A3 CON #TRI %C5. A sentence may contain a maximum of 31 characters.

A "statement" is a sentence which has been determined by the program to be either a given assumption input by the user; a conclusion derived by the program from any combination of previous statements; or a sentence possessing a reflexive property, whether input by the user or derived by the program. A statement must contain a consistent set, relation, set subsequence. The previous example of a sentence does not qualify as a statement, but SEG A3 CON SEG %C may.

A "conclusion" is a potential statement awaiting selection by the user or the program.

A "reason" is a character string which identifies the definition, property, or postulate of Euclidian Geometry used by the program to logically justify a statement. A reason may contain one to three three-character fields, followed by STA, followed by the statement numbers of the previous statements which imply the statement being

Thomas J. Kelanic, Taylor Allderdice High School, 2409 Shady Ave., Pittsburgh, PA 15217.

justified. An example of justification by the Transitive Property of Congruence of statement number seven, which is implied by statements one and two, is the following reason: 7. TRA PRO CON STA 1 2.

A "proof" is a sequence of statements, supported by reasons, which leads, by logical implication, to a desired concluding statement.

Implication Functions

An "implication function" is a portion of a computer program which applies tests to one or two statements and may then synthesize a resultant conclusion and supporting reason. The Symmetric Implication Function, for example, can operate on the statement numbered three:

3. SEG AB CON SEG XY

to produce the conclusion:

SEG XY CON SEG AB

and the reason:

SYM PRO CON STA 3

The Transitive Implication Function, for example, can operate on statements two and five:

2. #SEG AB = #SEG CD
5. #SEG CD = #SEG DE

to produce the conclusion:

#SEG AB =SEG DE

and the reason:

TRA PRO = STA 2 5

Hint Functions

To prove that two triangles are congruent by the A.S.A. Postulate of Plane Geometry, three statements are required. Two of these three state congruence of pairs of corresponding angles, and one states congruence of the pair of corresponding included sides. Instead of using a three-input implication function, the program uses a "hint function" and a single-input implication function. Of the three input statements required for the conclusion, any two are independent, and one is dependent. In other words, any one of the three required inputs is uniquely determined by the other two! The uniquely determined third statement is called a "hint," and the function producing it is called a "hint function." Hints are so-named because they may be output to the user, or used by the program itself, as a means of guidance. The single-input implication function which operates on hints to produce the conclusion of three required statements is called "The Hint-Implication Function." For example, the S.A. Hint Function can operate on statements one and two:

1. ANG ABC CON ANG DEF
2. SEG AB CON SEG DE

to produce two hints, two tentative conclusions, and two tentative reasons:

hint #1:	SEG BC CON SEG EF
tentative conclusion #1:	TRI ABC CON TRI DEF
tentative reason #1:	SAS STA 1 2
hint #2:	ANG BAC CON ANG EDF
tentative conclusion #2:	TRI ABC CON TRI DEF
tentative reason #2:	ASA STA 1 2

Later on, The Hint-Implication Function can operate on a statement such as number five:

5. ANG BAC CON ANG EDF

to produce

conclusion : TRI ABC CON TRI DEF
reason : ASA STA 1 2 5

or on statement number four:

4. SEG BC CON SEG EF

to produce:

conclusion: TRI ABC CON TRI DEF
reason: SAS STA 1 2 4

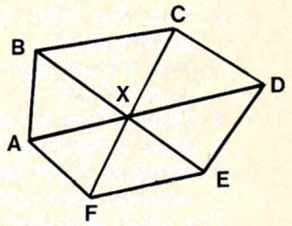
The Hint-Implication Function does not permit redundancy. If statement number four does result in the above conclusion, then all six outputs from the S.A. Hint Function are erased from the file PARTRUE.

Program Segment GEOMTREE1

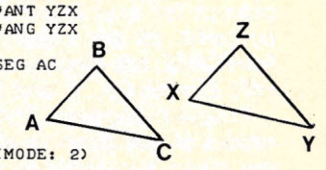
Program segment GEOMTREE1 comprises 325 lines of Extended BASIC, and contains the following functions, among others:

1. Single-Input Implication Functions:
   A. The Hint-Implication Function
   B. The Definition of Congruent Triangles
   C. The Symmetric Properties of Congruence and Equality
2. Dual-Input Implication Functions:
   A. The Transitive Properties of Congruence and Equality
3. Dual-Input Hint Functions:
   A. The A.A. Hint Function (for A.S.A.)
   B. The S.A. Hint Function (for A.S.A. and S.A.S.)
   C. The S.S. Hint Function (for S.A.S. and S.S.S.)

```
END AT 2510
*RUN
INPUT MODE 1,2,3, OR 4 ? 1
INPUT "GIVEN" G1 ? ANG XAB CON ANG FAX
INPUT "GIVEN" G2 ? #ANG BXA = #ANG AXF
INPUT "GIVEN" G3 ? SEG BC CON SEG FE
INPUT "GIVEN" G4 ? ANG CBX CON ANG EFX
INPUT "GIVEN" G5 ? #SEG CD = #SEG ED
INPUT "GIVEN" G6 ? PROVE
INPUT "TO PROVE" ? #ANG CDX = #ANG XDE
STA 0 CON 0 HIN 0
STA 1 CON 1 HIN 0
STA 2 CON 2 HIN 1
STA 3 CON 3 HIN 0
STA 4 CON 7 HIN 0
STA 5 CON 8 HIN 0
STA 6 CON 9 HIN 2
STA 7 CON 10 HIN 0
STA 8 CON 14 HIN 0
STA 9 CON 15 HIN 2
STA 10 CON 15 HIN 4
STA 11 CON 15 HIN 4
STA 12 CON 15 HIN 4
STA 13 CON 15 HIN 8
STA 14 CON 16 HIN 8
STA 15 CON 20 HIN 8
```




```
------------------------------------------------
GIVEN: G1: ANG BAX CON ANG FAX
       G2: #ANG AXB = #ANG AXF
       G3: SEG BC CON SEG EF
       G4: ANG CBX CON ANG EFX
       G5: #SEG CD = #SEG DE
PROVE:===> #ANG CDX = #ANG EDX  (MODE: .1)
------------------------------------------------
   STATEMENTS:                REASONS:
------------------------------------------------
1. #ANG BAX = #ANG FAX     1. GIV G1
2. #ANG AXB = #ANG AXF     2. GIV G2
3. #SEG AX = #SEG AX       3. REF PRO = STA 3
4. TRI ABX CON TRI AFX     4. ASA STA 1 2 3
5. #SEG BC = #SEG EF       5. GIV G3
6. #ANG CBX = #ANG EFX     6. GIV G4
7. #SEG BX = #SEG FX       7. DEF CON TRI STA 4
8. TRI BCX CON TRI FEX     8. SAS STA 5 6 7
9. #SEG CD = #SEG DE       9. GIV G5
10. #SEG AB = #SEG AF      10. DEF CON TRI STA 4
11. #ANG ABX = #ANG AFX    11. DEF CON TRI STA 4
12. #TRI ABX = #TRI AFX    12. DEF CON TRI STA 4
13. #SEG CX = #SEG EX      13. DEF CON TRI STA 8
14. #SEG DX = #SEG DX      14. REF PRO = STA 14
15. TRI CDX CON TRI EDX    15. SSS STA 9 13 14
16. #ANG CDX = #ANG EDX    16. DEF CON TRI STA 15
------------------------------------------------
Q.E.D.  EUCLID (3/24/1978)
------------------------------------------------
61 CPU USED.

 RUN
INPUT MODE 1,2,3, OR 4 ? 2
INPUT "GIVEN" G1 ? #SEG BC = #SEG ZY
INPUT "GIVEN" G2 ? #SEG BA = #SEG ZX
INPUT "GIVEN" G3 ? #ANG CBA = #ANT YZX
INPUT "GIVEN" G3 ? #ANG CBA = #ANG YZX
INPUT "GIVEN" G4 ? PROVE
INPUT "TO PROVE" ? SEG XY CON SEG AC
```




```
GIVEN: G1: #SEG BC = #SEG YZ
       G2: #SEG AB = #SEG XZ
       G3: #ANG ABC = #ANG XZY
PROVE:===> SEG XY CON SEG AC  (MODE: 2)
STA 0 CON 0 HIN 0
INPUT STATEMENT 1 ? G2

1. SEG AB CON SEG XZ                    1. GIV G2

NEW CONCLUSIONS FOUND:
C1: SEG XZ CON SEG AB
STA 1 CON 1 HIN 0
```

The over-all logic of GEOMTREE1 may be viewed under failure conditions. Failure of the examination of sets by 2A, above, leads to 3B Failure of the examination of identifiers by 2A leads to 3A or 3C. Hints, tentative conclusions, and tentative reasons are stored on file PARTRUE. Conclusions and reasons are stored on file CONFILE.

Program Segment—EUCLID

Program segment EUCLID contains the general logic of interactive proofs, a Turing Controller, and a User Command Interpreter. This program segment comprises 353 lines of Extended BASIC.

The general logic contained in EUCLID determines whether or not input from the user or the Turing Controller qualifies as a statement, as previously defined. Given assumptions, statements, and reasons are stored on the file PROFILE by EUCLID, for subsequent output as a finished proof.

The Turing Controller section of EUCLID provides for fully automatic theorem-proving by determining which step to take next. The Turing Controller admits defeat with the message EUCLID HAS FAILED, only after every valid possibility has been tried, but the final state, reaching the desired conclusion, has not been attained. The Turing Controller asks the following questions, where the word "match" is used to mean "identically equal to":

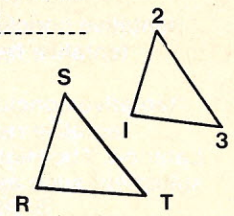
1. Does any Conclusion match the desired conclusion?
2. Has the Hint-Implication Function produced a conclusion?
3. Has a Dual-Input Implication Function produced a conclusion?
4. Does a hint possess a reflexive property?
5. Does a hint match a conclusion?
6. Have all given assumptions been used?
7. Have all conclusions been used?

The User Command Interpreter operates on all input strings having fewer than four characters. Commands beginning with a G or a C and immediately followed by one or two digits tell EUCLID which given assumption or conclusion has been selected for use as the next statement. For example, G5 or C10 are user commands indicating selection of the fifth given assumption or the tenth conclusion. Other user commands are used to output partial contents of the three files for review purposes. The command GIV results in output of the given assumptions stored, one per record, on the zeroth through n-1 records of PROFILE, where n is the number of given assumptions. The command STA results in output of a list of the statements made thus far in the proof, and stored on the *n*th, n+2, n+4, ... records of PROFILE. The command PRO results in output of the statement to be proved, located on record number three of CONFILE. The first six records of CONFILE are used for data storage and transmission from one program segment to the other. The command CON results in output of the remaining conclusions stored on the even-numbered records of CONFILE, beginning at record six. The command HIN results in output of the remaining hints stored on every third record of PARTRUE, beginning at record zero. To terminate program execution, the user may input END or hit the ESC key. Immediately prior to each program-prompt for user input of a new statement, the available contents of the three files PROFILE, CONFILE, and PARTRUE are displayed by the message STA X CON Y HIN Z, where X, Y, and Z are the appropriate numbers.

Program segment EUCLID offers the user one of four modes of operation. In all four modes, the user must input the given assumptions and the statement to be proved. The four modes are as follows:

1. *Demonstration Mode*. The Turing Controller directs EUCLID in an attempt to prove the theorem automatically. The program supplies all statements, reasons,

```
INPUT STATEMENT 2 ? G2
REDUNDANT SEE STA 1
INPUT STATEMENT 2 ? G1

2. SEG BC CON SEG YZ                    2. GIV G1

NEW CONCLUSIONS FOUND:
C2: SEG YZ CON SEG BC
STA 2 CON 2 HIN 2
INPUT STATEMENT 3 ? HIN

HINTS:
ANG ABC CON ANG XZY ?
SEG AC CON SEG XY ?
INPUT STATEMENT 3 ? G3

3. ANG ABC CON ANG XZY                  3. GIV G3

NEW CONCLUSIONS FOUND:
C3: TRI ABC CON TRI XZY
C4: ANG XZY CON ANG ABC
STA 3 CON 4 HIN Ø
INPUT STATEMENT 4 ? C3

4. TRI ABC CON TRI XZY                  4. SAS STA 1 2 3

NEW CONCLUSIONS FOUND:
C5; TRI XYZ CON TRI ACB
C6; SEG AC CON SEG XY
C7; ANG BAC CON ANG YXZ
C8; ANG ACB CON ANG XYZ
C9: #TRI ABC = #TRI XZY
STA 4 CON 8 HIN Ø
INPUT STATEMENT 5 ? C3
INVALID CODE
INPUT STATEMENT 5 ? PRO
PROVE:===> SEG XY CON SEG AC  (MODE: 2)
INPUT STATEMENT 5 ? SEG AC CON SEG XY

5. SEG AC CON SEG XY                    5. DEF CON TRI STA 4

NEW CONCLUSIONS FOUND:
C1Ø: SEG XY CON SEG AC
STA 5 CON 8 HIN Ø
INPUT STATEMENT 6 ? CON

CONCLUSIONS:
C1: SEG XZ CON SEG AB
C2; SEG YZ CON SEG BC
C4; ANG XZY CON ANG ABC
C5: TRI XYZ CON TRI ACB
C7: ANG BAC CON ANG YXZ
C8: ANG ACB CON ANG XYZ
C9: #TRI ABC = #TRI XZY
C1Ø: SEG XY CON SEG AC
INPUT STATEMENT 6 ? C1Ø

6. SEG XY CON SEG AC                    6. SYM PRO CON STA 5

INPUT YOUR FULL NAME ? THOMAS J. KELANIC
---------------------------------------------
GIVEN: G1: #SEG BC = #SEG YZ
       G2: #SEG AB = #SEG XZ
       G3: #ANG ABC = #ANG XZY
PROVE:===> SEG XY CON SEG AC  (MODE: 2)
---------------------------------------------
   STATEMENTS:               REASONS:
---------------------------------------------
1. SEG AB CON SEG XZ      1. GIV G2
2. SEG BC CON SEG YZ      2. GIV G1
3. ANG ABC CON ANG XZY    3. GIV G3
4. TRI ABC CON TRI XZY    4. SAS STA.1 2 3
5. SEG AC CON SEG XY      5. DEF CON TRI STA 4
6. SEG XY CON SEG AC      6. SYM PRO CON STA 5
---------------------------------------------
Q.E.D.  THOMAS J. KELANIC  (3/24/1978)
---------------------------------------------
53 CPU USED.

END AT 251Ø
*RUN
INPUT MODE 1,2,3, OR 4 ? 3
INPUT "GIVEN" G1 ? SEG RS CON SEG 21
INPUT "GIVEN" G2 ? SEG  ST CON SEG 32
INPUT "GIVEN" G2 ? SEG ST CON SEG 32
INPUT "GIVEN" G3 ? SEG 13 CON SEG RT
INPUT "GIVEN" G4 ? PROVE
INPUT "TO PROVE" ? TRI TSR CON TRI 321

GIVEN: G1: SEG RS CON SEG 12
       G2; SEG ST CON SEG 23
       G3: SEG 13 CON SEG RT
PROVE:===> TRI RST CON TRI 123  (MODE: 3)
STA Ø CON Ø HIN Ø
INPUT STATEMENT 1 ? G1

1. SEG RS CON SEG 12                    1. GIV G1 (O.K)

NEW CONCLUSIONS FOUND:
C1: SEG 12 CON SEG RS
STA 1 CON 1 HIN Ø
INPUT STATEMENT 2 ? G2

2. SEG ST CON SEG 23                    2. GIV G2 (O.K)
```



conclusions, and hints to itself. The user sits back and watches the changing file status message.

2. *Practice Mode.* EUCLID supplies all reasons, all conclusions, and permits user access to hints and conclusions. The user must supply all statements.

3. *Quiz Mode.* EUCLID supplies all conclusions, checks user reasons, and permits user access to hints and conclusions. The user must supply all statements and all reasons.

4. *Test Mode.* EUCLID checks the reasons. User access to hints, conclusions, and file status is denied. The user must supply all statements, reasons, and conclusions.

Also contained in EUCLID is The Implicit Definition of Congruence, which automatically selects the proper subsequence of set, relation, and set for each statement of congruence or equality, based on the relation found in the statement to be proved. If the user wishes to conclude with a statement of equality, for example, all statements of congruence, except for congruent triangles, are automatically changed to statements of equality. No reason is given for implicit definitions, but it simplifies EUCLID's task.

Future Development

Both program segments can easily be modified to accept other sets such as SGM, NGL, and TRN, instead of SEG, ANG, and TRI, since only one line in each program segment contains sentence vocabulary. This means, of course, that EUCLID is potentially multi-lingual. Additional relations, such as +, -, *, /, >, and < will be used in additional GEOMTREES. Other sentence sequences are presently acceptable to the program, and will be used in additional GEOMTREES. The new GEOMTREES will contain additional definitions, properties, and postulates from Plane Geometry. Eventually, it will be possible to prove practically all of the approximately three-hundred or so theorems usually found in the traditional high-school geometry course. It is not known if indirect proofs can be included, but probably so. In the future, ALGTREES will be written and linked to EUCLID for use in equation-solving. A student might issue the command D5, for example, to see the result of dividing each term of an equation by five. Also in the future, TRIGTREES will be linked to EUCLID for use in proving trigonometric identities. Will there be a CALCTREE? A BOOLTREE? A TREETREE? Compatible graphics will be developed for use with EUCLID. The graphics display will show a blank screen, initially. As statements are made in a proof, labeled points and segments will be added to the display. The segments will be stretched, shrunken, rotated, and translated to produce a representation of the relationships established in the proof. The evolving diagram will reach a final state that depicts all relationships established by the completed proof, as well as those relationships not established.

Copies of the program listing are available from the author at the following address:

Thomas J. Kelanic
Taylor Allderdice High School
2409 Shady Ave.
Pittsburgh, PA 15217

Requests must be accompanied by a check in the amount of $2.00 and payable to "Taylor Allderdice High School" to cover costs of copying and first-class postage.

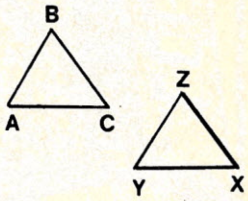
Bibliography

"Extended BASIC Users Manual," 093-000065-06, Data General Corp., rev. 06, February, 1975.

Kelanic, Thomas J. "Geometric Proofs," *Creative Computing,* Vol. 2, No. 2. Mar-Apr, 1976, 60-1.

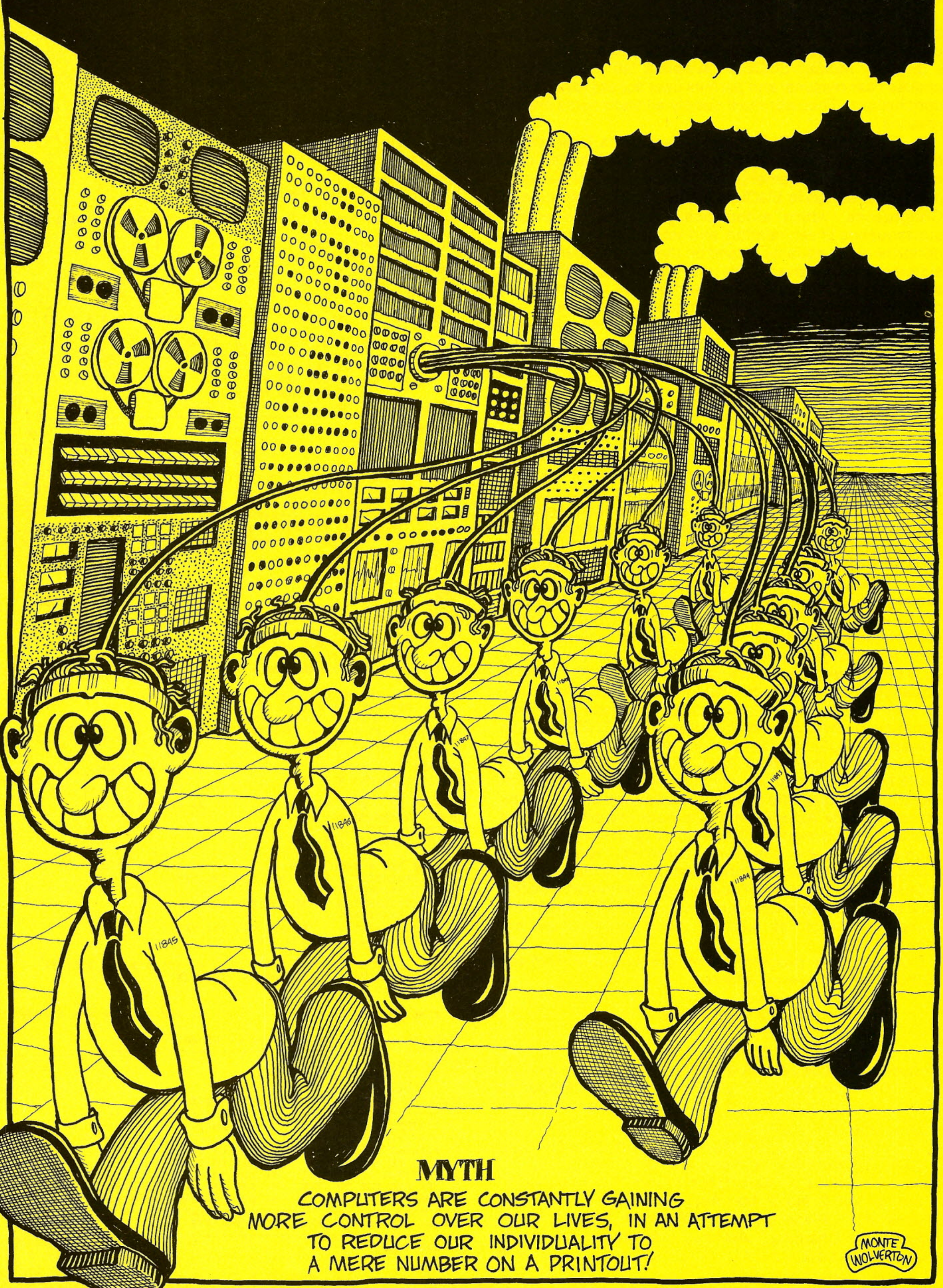
Nevins, A.J. "Plane Geometry Theorem Proving Using Foreward Chaining," *Artificial Intelligence.* VI. No 1 Jan, 1975, 1-23. ■


```
NEW CONCLUSIONS FOUND:
C2: SEG 23 CON SEG ST
STA 2 CON 2 HIN 2
INPUT STATEMENT 3 ? HIN

HINTS:
ANG RST CON ANG 123 ?
SEG RT CON SEG 13 ?
INPUT STATEMENT 3 ? SEG 13 CON SEG RT
INPUT REASON 3 ? G3

3. SEG 13 CON SEG RT                    3. GIV G3 (ERR)

NEW CONCLUSIONS FOUND:
C3: SEG RT CON SEG 13
STA 3 CON 3 HIN 2
INPUT STATEMENT 4 ? C3
INPUT REASON 4 ? SYM PRO CON STA 3

4. SEG RT CON SEG 13                    4. SYM PRO CON STA 3 (O.K)

NEW CONCLUSIONS FOUND:
C4: TRI RST CON TRI 123
STA 4 CON 3 HIN 0
INPUT STATEMENT 5 ? C4
INPUT REASON 5 ? SSS STA 1 2 4

5. TRI RST CON TRI 123                  5. SSS STA 1 2 4 (O.K)

INPUT YOUR FULL NAME ? THOMAS J. KELANIC
------------------------------------------------------------
GIVEN: G1: SEG RS CON SEG 12
       G2: SEG ST CON SEG 23
       G3: SEG 13 CON SEG RT
PROVE:===> TRI RST CON TRI 123  (MODE: 3)
------------------------------------------------------------
   STATEMENTS:              REASONS:
------------------------------------------------------------
1. SEG RS CON SEG 12     1. GIV G1 (O.K)
2. SEG ST CON SEG 23     2. GIV G2 (O.K)
3. SEG 13 CON SEG RT     3. GIV G3 (ERR)
4. SEG RT CON SEG 13     4. SYM PRO CON STA 3 (O.K)
5. TRI RST CON TRI 123   5. SSS STA 1 2 4 (O.K)
------------------------------------------------------------
Q.E.D.  THOMAS J. KELANIC  (3/24/1978)
------------------------------------------------------------
43 CPU USED.
```

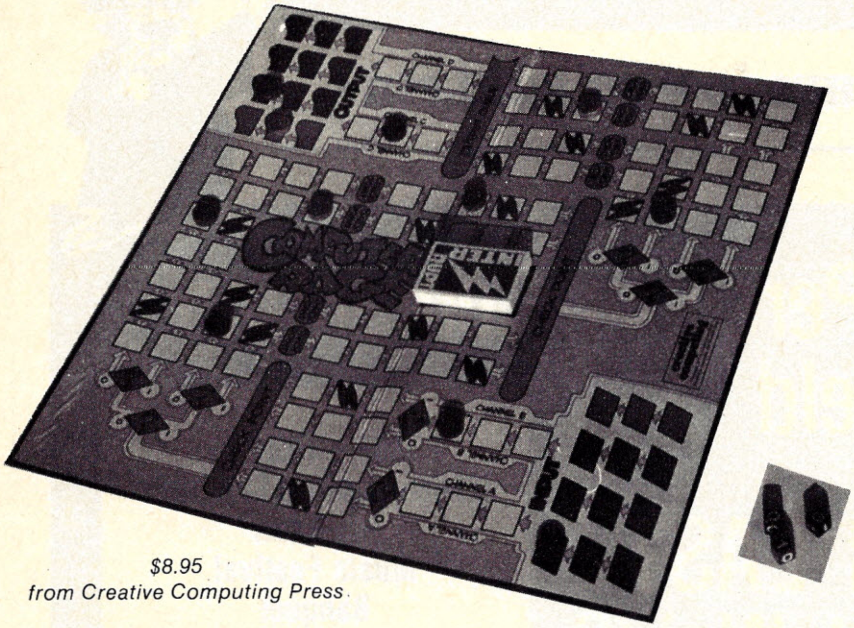



```
*RUN
INPUT MODE 1,2,3, OR 4 ? 4
INPUT "GIVEN" G1 ? SEG AC CON SEG YX
INPUT "GIVEN" G2 ? #SEG BA = #SEG YZ
INPUT "GIVEN" G3 ? ANG BAC CON ANG ZYX
INPUT "GIVEN" G4 ? PROVE
INPUT "TO PROVE" ? #TRI CBA = #TRI XZY

GIVEN: G1: SEG AC CON SEG XY
       G2; #SEG AB = #SEG YZ
       G3: ANG BAC CON ANG XYZ
PROVE:===> #TRI ABC = #TRI YZX  (MODE: 4)
INPUT STATEMENT 1 ? G1

1. #SEG AC = #SEG XY                    1. GIV G1 (O.K)
INPUT STATEMENT 2 ? G2

2. #SEG AB = #SEG YZ                    2. GIV G2 (O.K)
INPUT STATEMENT 3 ? G1
REDUNDANT SEE STA 1
INPUT STATEMENT 3 ? G3

3. #ANG BAC = #ANG XYZ                  3. GIV G3 (O.K)
INPUT STATEMENT 4 ? TRI ABC CON TRI XYZ
INVALID STATEMENT
INPUT STATEMENT 4 ? TRI ABC CON TRI XZY
INVALID STATEMENT
INPUT STATEMENT 4 ? TRI ABC CON TRI YZX
INPUT REASON 4 ? SAS STA 1 2 3

4. TRI ABC CON TRI YZX                  4. SAS STA 1 2 3 (O.K)
INPUT STATEMENT 5 ? #TRI ABC = #TRI YZX
INPUT REASON 5 ?

5. #TRI ABC = #TRI YZX                  5. DEF CON TRI STA 4 (ERR)

INPUT YOUR FULL NAME ? THOMAS J. KELANIC
------------------------------------------------------------
GIVEN: G1; SEG AC CON SEG XY
       G2; #SEG AB = #SEG YZ
       G3: ANG BAC CON ANG XYZ
PROVE:===> #TRI ABC = #TRI YZX  (MODE: 4)
------------------------------------------------------------
   STATEMENTS:              REASONS:
------------------------------------------------------------
1. #SEG AC = #SEG XY     1. GIV G1 (O.K)
2. #SEG AB = #SEG YZ     2. GIV G2 (O.K)
3. #ANG BAC = #ANG XYZ   3. GIV G3 (O.K)
4. TRI ABC CON TRI YZX   4. SAS STA 1 2 3 (O.K)
5. #TRI ABC = #TRI YZX   5. DEF CON TRI STA 4 (ERR)
------------------------------------------------------------
Q.E.D.  THOMAS J. KELANIC  (3/24/1978)
------------------------------------------------------------
35 CPU USED.
```

COMPUTER MYTHS EXPLAINED



creative computing



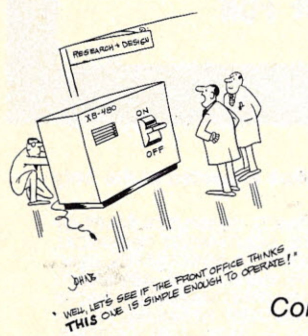
Computer Rage

This fun and educational new board game is based on a large-scale multiprocessing computer system. The object is to move your three programs from input to output. Moves are determined by the roll of three binary dice representing bits in a computer. Hazards include priority interrupts, program bugs, decision symbols, power failures and restricted input and output channels. Notes are included for adapting game for school instruction. A perfect introductory tool to binary math and the seemingly-complex computer. [6Z]

$8.95
from Creative Computing Press

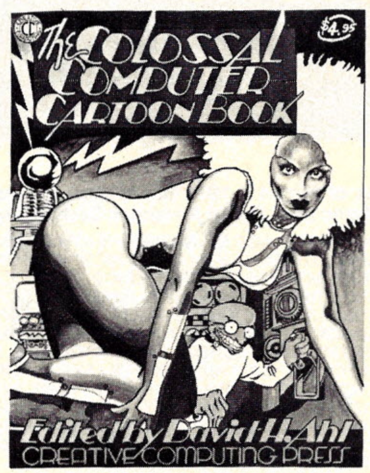
Binary Dice

Now, the same dice used in Computer Rage can be purchased separately. Three binary dice (red, green and blue) in a zip-lock bag. $1.25 postpaid [3G].

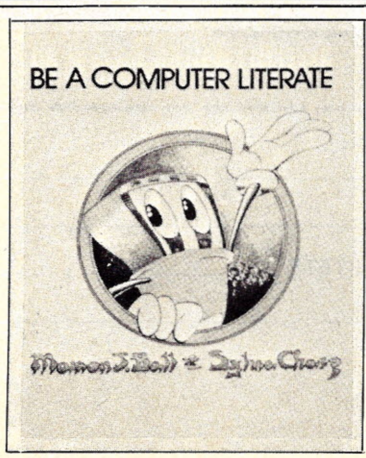


The Colossal Computer Cartoon Book

Take a break. Sit back and and relax with the biggest and best collection of computer cartoons ever, hundreds and hundreds of cartoons about computers, robots, calculators AI and much more. [6G]



$4.95
120 pp. softbound
from Creative Computing Press



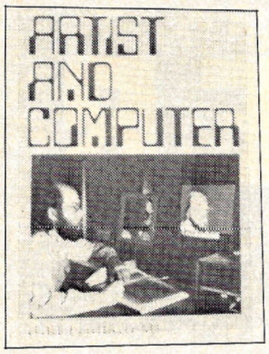
$3.95
61 pp. softbound
from Creative Computing Press

Be A Computer Literate

This is the most basic, introductory book on computers ever put together for instructional use. Its full-color diagrams, drawings, photos and large, explicit type make this book a pleasure to read. This chapter titles, themselves, best illustrate its contents— [6H]

I Introduction
II What Are Computers
III Kinds of Computers
IV What Goes On Inside Computers
V Communicating With The Computer
VI Language Of The Computer
VII How To Write A Simple Program
VIII How Computers Work For Us
Glossary

brings you its best



Artist and Computer

"Get yourself a copy of this book if you enjoy feeding your mind a diet of tantalizing high-impact information."
San Francisco Review of Books.

$4.95
121 pp. softbound
from Creative Computing Press

This unique art book covers a multitude of computer uses and the very latest techniques in computer-generated art. In its pages, 35 artists explain how the computer can be programmed either to actualize the artist's concept (such as the visualization of fabric before it is woven) or to produce finished pieces. Over 160 examples, some in full color. [6D]

The Best of BYTE

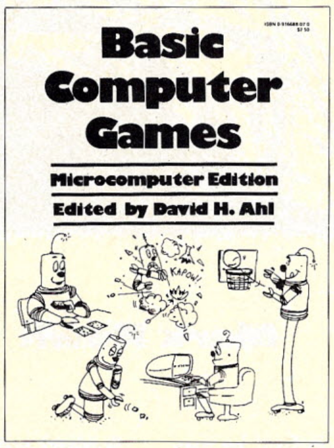
$11.95
386 pp. softbound
from Creative Computing Press

This is a blockbuster of a book containing the majority of material from the first 12 issues of *Byte* magazine. The 146 pages devoted to hardware are crammed full of how-to articles on everything from TV displays to joysticks to cassette interfaces and computer kits. But hardware without software might as well be a boat anchor, so there are 125 pages of software and applications ranging from on-line debuggers to games to a complete small business accounting system. A section on theory examines the how and why behind the circuits and programs, and "opinion" looks at where this explosive new hobby is heading. [6F]

Basic Computer Games: Microcomputer Edition

New revised edition of our most popular book, *101 BASIC Computer Games.* All you need is a basic-speaking computer.

$7.50
185 pp. softbound
from Creative Computing Press



Here are 102 classic computer games, every one in standard microcomputer BASIC. Every one is complete with large legible listing, sample run and descriptive notes.

All the classics are here: Super Star Trek (one of the most challenging versions anywhere), Football (two versions), Blackjack, Lunar Lander (three versions), Tic Tac Toe, Nim, Life and Horserace—to name a few.

Guessing games, matrix games, word games, plotting games, card games, educational games—they're all here. And, they'll all run on your Altair, Imsai, Radio Shack, SWTPC, Xitan, OSI, Poly, Sol, PDP-11 or other micro or mini with extended BASIC.

The delightful cartoons on every page, coupled with highly legible listings, make this revision of *101 BASIC Computer Games* a real must, even if you own the original. [6C]

Volume 1

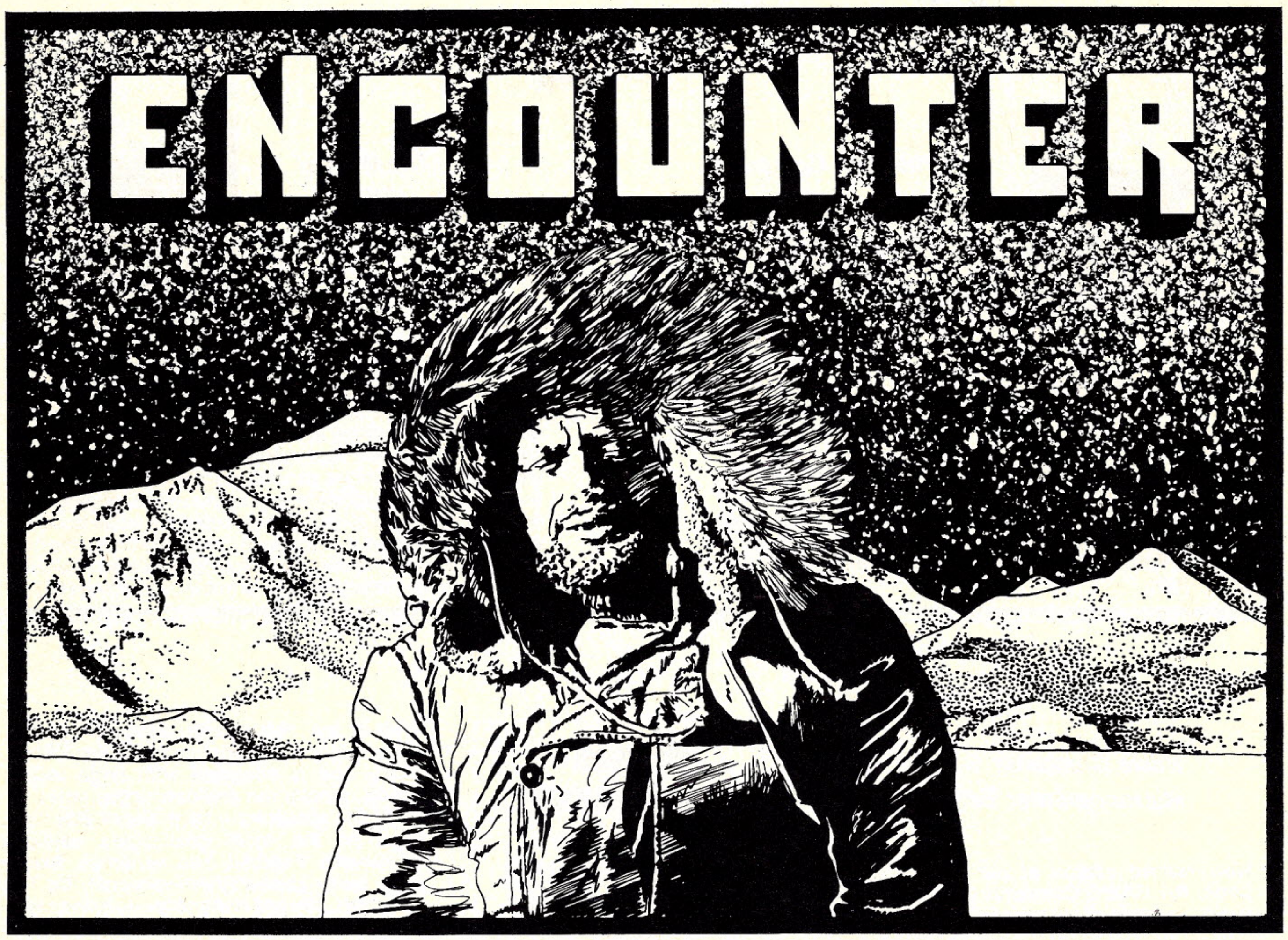
$8.95
328 pp. softbound
from Creative Computing Press

The first two years of *Creative Computing* magazine have been edited into two big blockbuster books. *American Vocational Journal* said of Volume 1, "This book is the 'Whole Earth Catalog' of computers." [6A] Volume 2 continues in the same tradition. "Non-technical in approach, its pages are filled with information, articles, games and activities. Fun layout."—*American Libraries.* [6B]

Volume 2

$8.95
336 pp. softbound
from Creative Computing Press

To order call toll-free
800-631-8112
(in NJ call 201-540-0445)
fill in the inserted order card or write to:
creative computing
Attn: Marie, P.O. Box 789-M, Morristown, NJ 07960.
*All book orders must be prepaid. Include $1 for shipping, USA; $2, foreign.



Mark Rowh

Henderson was lost. He hadn't strayed far; in fact, he was probably less than three kilometers from the station. But in the wasteland which was the Antarctic, the distance might as well have been ten times as great. Unless he proceeded in exactly the right direction, he would miss the station completely. The snow was blowing wildly about him, and the temperature was already beginning to drop. Henderson knew that he must find the base within the next few hours, or his chances were nil. He would collapse and die, a frozen speck in the unbroken southern wastes, another victim of this hostile, barren land.

He peered into the swirling whiteness, his eyes burning, and cursed himself. He had broken the cardinal rule for those who dared to take their research past the 60th parallel: he had strayed, alone and without equipment, from the immediate environs of the research station. Now, only a few minutes later, he was beginning to pay. The cold was digging in relentlessly, gripping his body like a giant frozen fist. He had not meant to go far; he had simply wanted to walk, to think, to get away from the sweat and the alcohol and the mindless chatter of the others. His theory was beginning to crystallize, and he needed solitude to think. It was all starting to come together, like the tectonic plates upon which his theory rested. Soon, his work would be finished and he could return to civilization, where he would present the first definitive look at the geological history of the Antarctic Basin. He was so engrossed in his thoughts, though, that he had paid ilttle attention to where he was going; and when the weather changed, he was suddenly and dangerously lost.

He began to panic; he could see only a few meters, could hear nothing but his own labored breathing and the constant shrieking of the wind. "God," he said half-aloud, "Help me. Help me!" He screamed it this time, as loudly as he could, and then repeated himself, once, twice, thrice, plaintive appeals which no one would ever hear. As the last scream died away, he felt his strength ebb, as though it had been carried off with the sound of his voice, rapidly dissipating in the wind. His clenched fists relaxed, and desperate tears streamed down his face and froze. He began to sob, like a lost, forgotten child.

And then, at the very edge of his field of vision, he saw a light. It seemed a gift from heaven, a divine response to his pleas, the deus ex machina which would ring the curtain on an all too frightening act. He ran, flailing his arms and giggling like an idiot, relieved of the awful burden of facing death. It seemed a long time, but finally he reached the light. And then he felt as though lifted to a great height and dropped like a stone. For it was not

And then, at the very edge of his field of vision, he saw a light. It seemed a gift from heaven, a divine response to his pleas....

the station.

Instead it was only a small, blue, conical object, glowly steadily with an icy, penetrating light. It just rested there, silently and incongruously, and was like nothing Henderson had ever seen. His mind grappled, the scientist in him attempting to assess the situation, the man trying to overcome his fright.

Whatever it was, it seemed to be the tip of something much larger buried in the snow and ice. Henderson fell to his knees and began scraping, trying to uncover the entire object. If he were lucky, it might just be large enough to get inside, to provide some degree of shelter until the weather changed or the others came to find him. He dug furiously, the object looming larger as he worked. Gingerly, he touched it with an index finger. It was cold, smooth, and without any readily apparent energy field. Curious, he placed both palms against it and slid them down, pushing away the snow. Then, suddenly, like the plug of an ancient volcano, his mind exploded.

His brain was an expanding universe, crowded with onrushing galaxies of information, flooded with signs and sounds and screaming lights which echoed endlessly. They all seemed directed at him; and yet he could not understand, could not although his mind was straining to comprehend, and each atom of his being was glowing red with pain. Then, slowly, the red began to come together, to coalesce like molten wax, and gradually the pain receded. He felt that someone was talking to him, but the voices, in a strange inhuman tongue, seemed lost in the distance. Then, finally, a word came through, a word he could understand.

"Contact."

"Wh...what?"

"Biological, mammalian, biped. Marginally intelligent.

Harmless." The words did not seem directed at him, but about him. They seemed to echo metallically within his head, and he was not sure whether he heard or merely felt them. The catalog continued, item after item which apparently comprised the most detailed description of homo sapiens which he had ever encountered. Most of the descriptions were numerical, and some obviously chemical analysis. Finally they slowed, then stopped. "At last."

"What . . . are you?" He expelled the words like broken teeth. There was a long pause; it seemed that his very words were being weighed, felt, tested. Then, in glowing pulsations within his brain, the response came.

"Language . . . inconsistent . . . difficult . . . additional contact needed . . . will attempt . . .," and then the lights grew outward, his skull becoming a single glowing nova, his consciousness swept away in an immense irradiated field. Then, after an immeasurable span of time, his self-awareness returned.

"I . . . am . . . not designated. Not biological. I am . . . a . . . computer . . . but sentient, alive. I was formed . . . to monitor life activities . . . on this . . . continent." The words were beginning to flow together, to lose some of their halting quality, as though the thing were beginning to grasp the language, as though it had drawn knowledge directly from the man's memory.

Henderson, faced with this startling communication, was again the scientist. His mind began to fill with questions.

"But there is virtually no life here. And—who formed you? How did you get here?"

"You are . . . correct. Life is limited here. But life once . . . flourished. There was much data to collect."

"That was thousands of years ago!"

"Yes . . . long ago. I was placed here from . . . another world . . . very distant. I functioned . . . but then, climactic aberrations occurred. Life forms . . . expired . . . or were driven away. You are like . . . those creatures . . . your life forces are . . . fluctuating. The cold is . . . damaging."

"Yes! I must find shelter. Can you help me?"

There was a short silence. Henderson again became aware of the cruel sound of the wind.

"It is possible . . . to direct you . . . to the others of your species. The distance is not great. I can detect their presence. I have known you were here . . . have waited . . . for you to come."

He felt himself jump inside. The cold was penetrating, but he was confident he could make it back to the station, if only he knew in which direction to head.

"Please. In which direction should I go? That way?" He found himself pointing, although that probably meant nothing to the computer.

"I will not provide you . . . with that information."

"What? But I will die."

"I desire communication. It has been . . . has been . . . lonely. You will leave . . . and not come back. Those who made me . . . did not come back."

"I will! I will come back, I promise. I only need shelter, and warmth. Then I will return, will even bring others. There will be much communication."

"No. You are like those who placed me here. You will not return."

"But there will be no communication, for I cannot survive in this climate. I will die—will expire."

"It is not important. There is much in your mind. I will take it . . . and it will be as . . . nourishment . . . for a long time. Tell me . . . about your world."

"No! You must help me." But even as the words escaped from his rawing throat, Henderson felt his strength go. The wind was increasing its fury, and he could no longer feel his legs. He tried to stand, but could not move. The light in his brain was returning, the only spot of warmth in his body. "Please," he muttered, "Please."

"Interesting . . . quite interesting," the words formed.

He placed his hands against his temples, as though to block out the penetrating energy. But the effort was futile; there was nothing but the probing, all encompassing light.

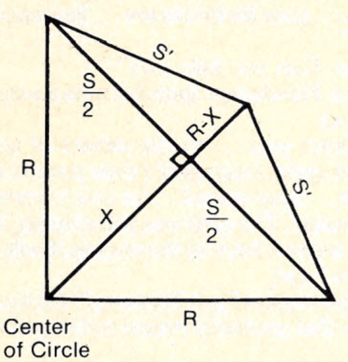
And it was growing. ■

..short programs...

Convergence Revisited

The "Convergence" Short Program on page 132 of the March-April 1978 issue of *Creative Computing* presented a method for calculating π by inscribed polygons. I have a program which doesn't use π to calculate π.

Since the circumference of a circle with radius R is $2\pi R$, we can approximate the circumference by inscribed polygons, dividing the perimeter by 2R to obtain an approximation to π. We can start with an inscribed square of side R*SQR(2) and double the number of sides for each calculation. If the old side length is S, then the length S' of a side of a new inscribed polygon with twice as many sides is obtained as follows:



Two applications of Pythagorus' Theorem yield the two equations

$$X^2 + (S/2)^2 = R^2$$

$$(R-X)^2 + (S/2)^2 = (S')^2$$

Thus

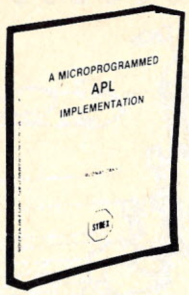
$$S' = \sqrt{(R-\sqrt{R^2-(S/2)^2})^2 + (S/2)^2}$$

It is easy to simplify this formula algebraically, but accuracy suffers if one does this. Also, one should use S*S instead of S**2, and SQR( ) instead of ( )**.5, although this may differ on various computers.

As the run shows, the calculation is accurate to 13 decimal places[compared with 4*ATN(1)], and actually differs only by $4*10^{-15}$. I ran it with the loop counter set at 30, but the value didn't change. I also experimented with different radii, and settled on R=10.

George W. Ball
Dept. of Mathematics
Alfred University
Alfred, NY 14802


```
LIST
100 REM PROGRAM TO CALCULATE PI BY INSCRIBED POLYGONS
110 REM GEORGE W. BALL
120 REM ALFRED UNIVERSITY
130 PRINT & PRINT
140 PRINT"THE MACHINE VALUE FOR PI IS";
150 PRINT PRC(1),4*ATN(1)
160 PRINT & PRINT
170 REM WE WILL USE A CIRCLE OF RADIUS 10
180 R=10
190 PRINT "SIDES","        PERIMETER"
200 PRINT
210 N=4,S=R*SQR(2)
220 FOR K=1 TO 28
230 PRINT PRC(1),N,(S*N)/(2*R)
240 Y=S*S/4
250 H=SQR(R*R-Y)
260 X=R-H
270 S=SQR(X*X+Y)
280 N=2*N
290 NEXT K
>RUN
11:09    MAR 09   RUNZBAA...


THE MACHINE VALUE FOR PI IS     3.141592653589793


SIDES                   PERIMETER

 4                  2.828427124746190
 8                  3.061467458920718
 16                 3.121445152258051
 32                 3.136548490545938
 64                 3.140331156954752
 128                3.141277250932772
 256                3.141513801144299
 512                3.141572940367089
 1024               3.141587725277158
 2048               3.141591421511198
 4096               3.141592345570116
 8192               3.141592576584870
 16384              3.141592634338560
 32768              3.141592648776983
 65536              3.141592652386589
 131072             3.141592653288990
 262144             3.141592653514590
 524288             3.141592653570990
 1048576            3.141592653585090
 2097152            3.141592653588615
 4194304            3.141592653589496
 8388608            3.141592653589716
 16777216           3.141592653589771
 33554432           3.141592653589785
 67108864           3.141592653589788
 134217728          3.141592653589789
 268435456          3.141592653589789
 536870912          3.141592653589789

          290 HALT
```



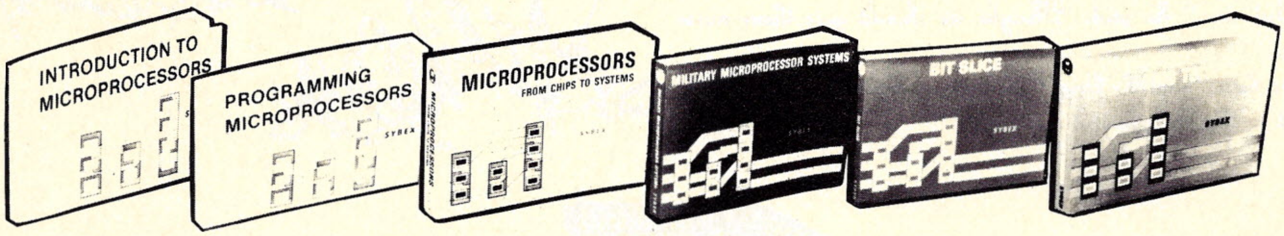
A MICROPROGRAMMED APL IMPLEMENTATION

RODNAY ZAKS, 350pp, ref. Z10 **$25.00**

The complete design and actual implementation of an APL interpreter, including the actual listing for the DSC Meta 4. Highly optimized, this very terse design resides in only 2K ROM locations. The book presents the complete theory and microprograms including n-dimentional arrays, memory management, n-level function nesting, APL operators, stack management, parsing.

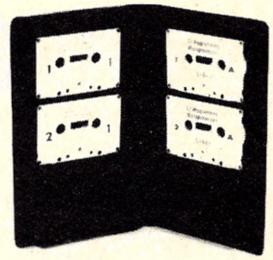
CONTENTS:

INTRODUCTION / PARSING APL AS A "NESTED PRIMITIVE LANGUAGE" / TRANSLATION FROM EXTERNAL TO INTERNAL REPRESENTATION / DYNAMIC APL STRUCTURES / THE INTERPRETER / THE OPERATORS / PERSPECTIVE / LISTINGS / TIMING.



SELF-STUDY COURSES

Each course includes a special book plus regular audio cassettes. Study at home, in the office, in your car.



1—INTRODUCTORY:

S1—INTRODUCTION TO MICROPROCESSORS (2½ hrs) **$29.95**
Basic introductory hardware course.

S2—PROGRAMMING MICROPROCESSORS (2½ hrs) **$29.95**
Basic programming and introduction execution at the MPU level.

2—INTRODUCTORY TO INTERMEDIATE

SB1—MICROPROCESSORS (12 hrs) **$59.95**
The most comprehensive course, from introductory concepts to programming, to assembling an actual system. Includes S1 and S2. 550pp ref. book.

3—SPECIALIZED

SB3—MILITARY MICROPROCESSOR SYSTEMS (6 HRS)................................ **$49.95**

SB5—BIT-SLICE: how to assemble a real CPU. IN detail. Other applications. **$49.95**

SB7—MICROPROCESSOR INTERFACING TECHNIQUES **$49.95**
From keyboard to floppy disk, including the standard buses S100, IEEE488, etc.)

ALSO AVAILABLE: VIDEOTAPES for SB1, SB3, SB5, SB7

☐ **MICROPROCESSORS,** RODNAY ZAKS, 420pp, ref. C201............ **$9.95**

☐ **MICROPROCESSOR INTERFACING TECHNIQUES,**
AUSTIN LESEA AND RODNAY ZAKS, 410pp, ref C207.................. **$9.95**

☐ **MICROPROCESSOR LEXICON...$2.95**
(Complete dictionary, 110pp)

DISTRIBUTION / TRANSLATION INQUIRIES INVITED

TO ORDER

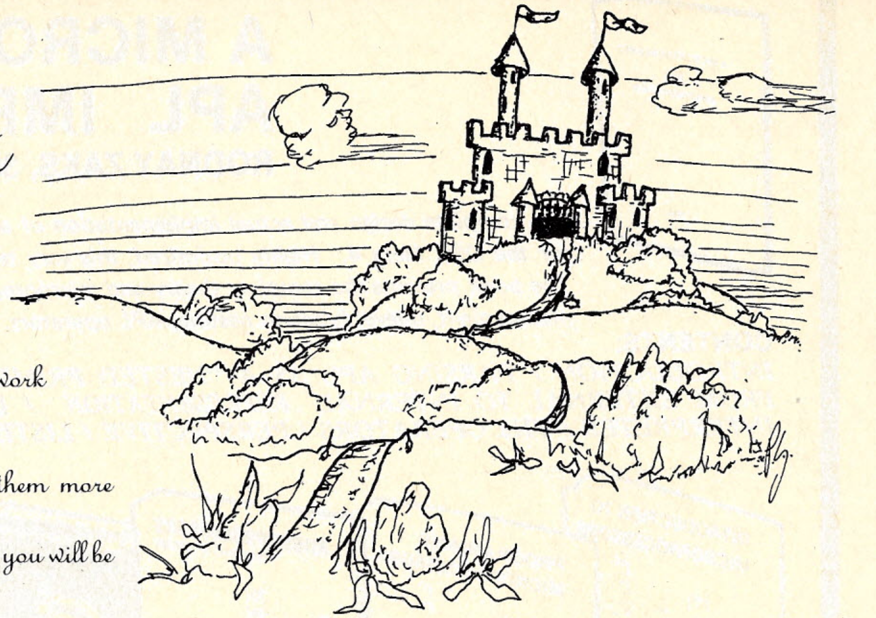
- **BY PHONE:** call (415) 848-8233 BankAmericard, Mastercharge accepted
- **SHIPPING:** free if payment included **ADD:** $1.00/book for fast shipping.
- **TAX:** in California, add sales tax.
- **OVERSEAS:** Write to: SYBEX-EUROPE, 313 rue Lecourbe 75015 - PARIS, France Tel:(1)8282502

SYBEX
2020 Milvia St.
Berkeley,
CA 94704
Tel: (415) 848-8233

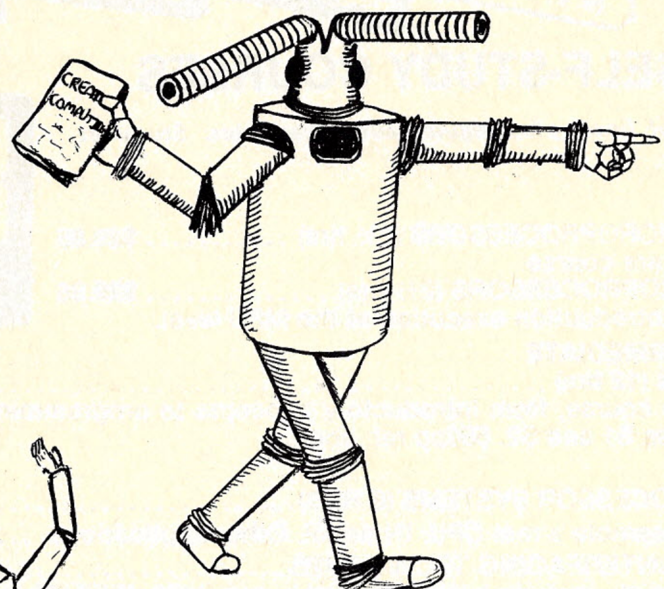
NAME__________ POSITION __________
COMPANY __________
ADDRESS __________
CITY__________ STATE/ZIP __________
☐ Z10 ☐S1 ☐S2 ☐B1 ☐B3 ☐B5 ☐B7 ☐Other________
☐ Payment enclosed ☐ Bill company ☐ Bill me
ADD $1.00 / BOOK FOR FAST SHIPPING
☐ charge my ☐ Visa ☐ Master charge
☐ Number__________ Exp date __________
Signature__________ ☐ Send catalog.

CC5/78

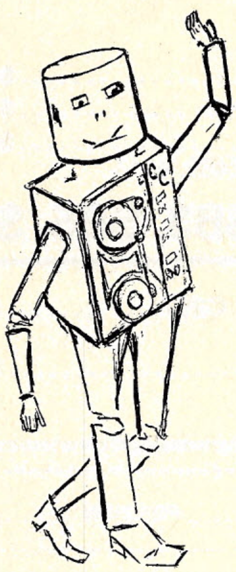
Cartoon Contest



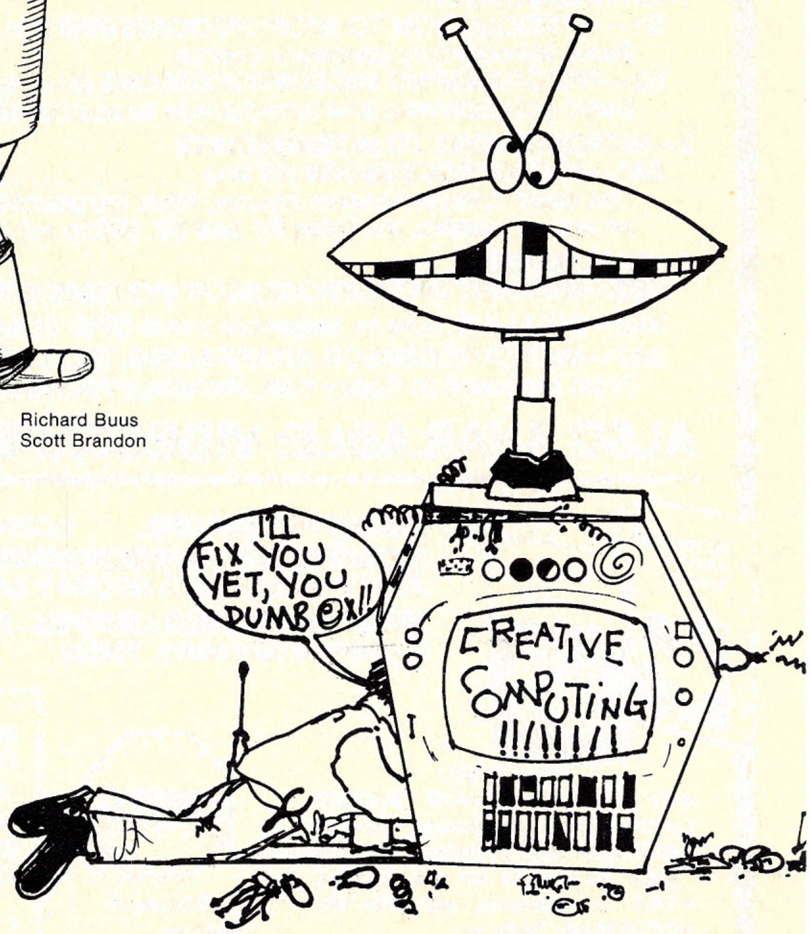
Once upon a time
In a program quite close to our own
There was a cartoon contest.
Many artists from far and wide sent their best work
But the "head honcho" was not satisfied.
At pencil-point, the art director came forth.
"But sire," he said, "Maybe we should give them more time."
"So be it," said the king, "But if you should fail, you will be sent to the reprogramming center."

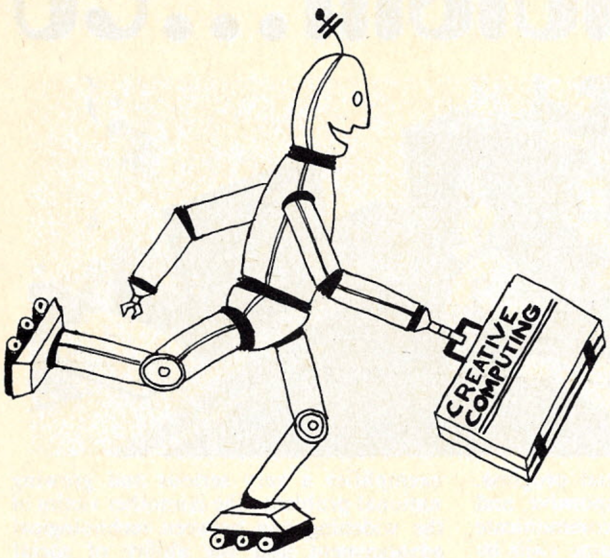


Richard Buus
Scott Brandon

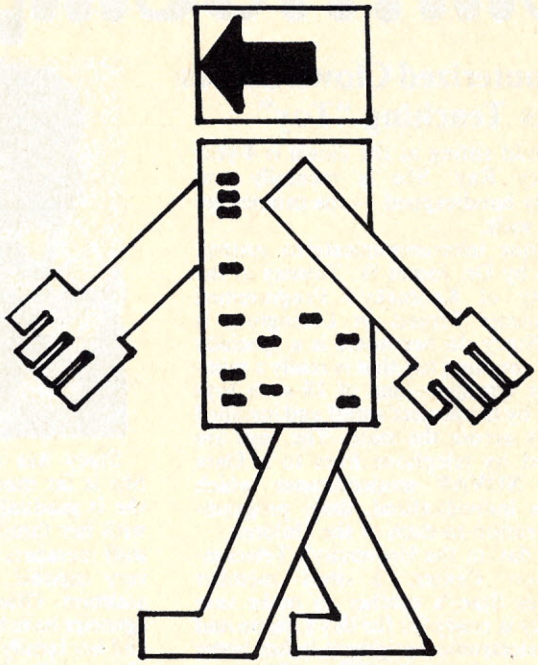


Dennis M. Sharp





David Jenks



David Daruszka

So, folks, help me out of a jam. Send in some more entries. I'm allergic to electrodes. Here's the input:

Be the *Creative* winner, or at least be a *Creative* contestant in our "Draw-A-Robot Contest."

We want all of you *Creative Computing* readers to try your hand at creating a robot. Not a mechanical demon or a lively cohort. Just a simple, modern, humanized version of our out-dated robot appearing on the contents page. We need a versatile robot to be part of our logo—one that symbolizes what *Creative* is all about.

The winning robot will appear on our advertisements, business cards, stationery, stickers, in the magazine—you name it. Who ever comes in contact with *Creative Computing* will come in contact with *the* robot.

Let your imagination run wild. Anything goes, but here's what we're looking for....

A robot that reveals character and progress, a trustworthy face and a strong physique. A robot in motion who carries an inherent sense of innovation. In other words, a robot that reflects *Creative Computing.*

The winner will receive $25 plus a biosketch which will accompany the featured full-page robot. The five runner-up robots will appear with the winner in the Nov-Dec issue.

Here are the specifications:

—All drawings must be in by Oct. 1, 1978.

—A separate sheet with your name, address and phone number must be attached to the drawing.

—Robot drawings larger than 9x12 or smaller than 1x3 will not be considered.

—Black on white paper only.

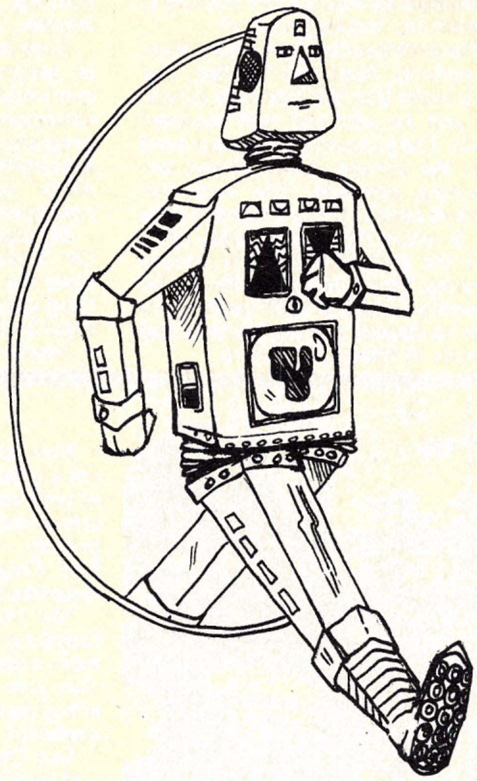
—There is no limit to the number of drawings you can submit.

—Send all robots to:

Draw-A-Robot Contest
Creative Computing
P.O. Box 789-M
Morristown, NJ 07960

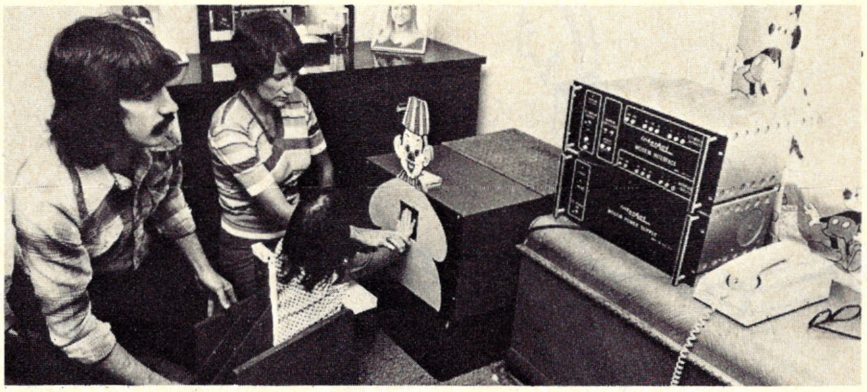
Here are some of the entries already submitted. Although clever, they're not exactly what we want, but they have a lot of merit.

P.g.



Lynette Nelson

Computerized Clown Serves As Teaching "Toy"



The child sitting at the clown is 4-year old Stacy Ray. She is mentally and physically handicapped — she can neither talk nor walk.

A unique telecommunications system, designed by Dr. James W. Tawney at the University of Kentucky's Programmed Environments Project in Lexington, is helping Stacy. In her home is a specially designed teaching toy that is really a mini-computer terminal—one of 18 such toys designed by the project's staff and installed in homes across the state. The toys are connected by telephone lines to a Data General NOVA® minicomputer which transmits individualized, daily programmed instrution to each of the children.

Every day in the University's Telecommunication Center, a staff member telephones Stacy's mother to make sure that Stacy is ready for the day's lesson and that the electronic equipment in her home is functioning property. When all systems are "go," the computer takes over. With her mother's assistance, Stacy starts her day's lesson.

Right now, Stacy's toy is teaching her to tell the difference between a picture of a rabbit and a picture of a cap. Learning to make visual discriminations like this one is a prerequisite for learning to read.

When Stacy makes the right choice from the two pictures displayed for her, her toy, which looks like the face of a clown, lights up and an attached tape recorder plays music. The pictures stay in place until she makes the correct choice in this no-error teaching program. The computer keeps track of each event and stores it in its memory. It plots Stacy's progress on a graph. At the end of the ten to twenty minute lesson, Stacy's mother talks on the phone directly with the staff operator in Lexington to discuss the day's lesson and other aspects of Stacy's condition.

Stacy has made substantial progress. She is far more alert and responsive, and she is gradually learning to communicate with her family. Frequent home visits by staff members have made Stacy's mother a very capable, highly motivated teaching assistant. Other health and social service agencies have been tapped to provide Stacy and her family with the support which has, along with her computerized learning programs, made Stacy's progress possible.

The story has an unhappy ending. The two years of federal funding which sponsored the project is almost all gone. The project was designed to find out if a telecommunications, computer-based learning program is technologically feasible. The results are clear: it is feasible.

Toys like Stacy's have also been placed in geographically isolated homes where conventional delivery of social services is extremely difficult and infrequent. Some programs promote muscular control for developmentally retarded children while other programs, like Stacy's, teach rudimentary cognitive skills.

By ingeniously stretching every dollar, Dr. Tawney's staff has managed to continue the program for a few families into a third year, but the remaining programs will soon be discontinued.

Stacy's dilemma is not unique. It exemplifies a very serious and growing national problem. She is another victim of the widening gap between technological advancement and the ability of social service agencies to deliver coordinated services. The technology exists; the service lags far behind. To help these children, systematic detection of multiple handicapped newborns and early intervention, starting in infancy, with validated curricula for both children and parents are both possible and necessary.

Starting from scratch, Dr. Tawney and his staff developed the prototype technology. They designed and implemented the entire telecommunications system for under $200,000 in less than two years. Tawney estimates that to mobilize existing agencies to implement the technology and the programs statewide will take more than a decade and will cost millions.

The annual cost per child of Tawney's computer-based telecommunications system is just under $2,900. Even adding in the costs of accompanying medical and social services, the full cost of supporting a severely handicapped child at home is still only a third of the cost of state institutionalization. As technological advances further reduce equipment costs, that gap will widen.

Bank Customers Operate Computer

Now, instead of having to stand in line to ask a teller for information about the status of your savings account or home loan, you can obtain such information — in complete privacy — from a computer terminal at 22 San Diego Federal Savings and Loan Association offices.

"By literally putting the customer in finger-tip touch with San Diego Federal's main computer, we can save him time and also give the teller more time to handle actual money transactions," said the bank's president.

San Diego Federal was the first savings and loan association in America to provide this service. Here's how it works.

The customer inserts a wallet-sized, plastic "Passcard" with a magnetically coded stripe on the back into an IBM 3606 desk-top terminal. The electronic stripe instantaneously tells the computer to get ready to answer questions about the particular account the Passcard represents.

Then, after the customer keys in the first four letters of his secret "Password" (known only to the customer and the computer), he can key in questions about his current savings balance, the last transaction date and amount, and interest earned in the current quarter for the current and previous year.

In addition, the read-out display on the "A$K" terminal also can tell the customer his current home loan balance, interest paid this quarter, in the current and previous year, and the amount and date of his next loan payment.

The terminal is limited to only providing information about an account. It cannot make entries to or change the account.

Computer Politics

Shortly before Edward Koch was elected Mayor of New York City, he was receiving daily poll results. What was not known at the time by many other political pros was that these poll results were obtained almost instantaneously on the same day they were taken. This breakthrough in political polling was devised by the firm of Penn & Schoen with the assistance of a personal computer system purchased at a local computer store.

Mark Penn, a partner of Penn & Schoen Associates of New York City, and a law student at Columbia University, knew there had to be a better way to analyze political polls. "It's ridiculous to wait a week while raw data is being keypunched and fed into a large computer to be analyzed," he said. "If there is a flaw in a question, or a poll taker doesn't understand what to do, you might not see the problem until a week of mistakes has occurred. All the results that follow have to be suspect until the problem has finally passed."

Penn felt that an interactive computer system could be used to monitor the poll and find problems in advance. Since political campaigns are notoriously strapped for cash, and the labor costs of poll taking are so high, the investment in a computer system was a major consideration. The solution, according to Penn, was a personal computer system. After a trip to The Computer Corner, a computer store in White Plains, Penn decided to buy a Sol 20 Computer with a North Star minidisk drive. "We chose the Sol 20 because it's a complete computer with a keyboard and printer output built in. The North Star disk, although low in storage capacity, had the lowest price for any disk system. It also had a good extended disk BASIC interpreter," said Penn. "Later on we purchased a used Centronics 102A character printer, North Star floating point arithmetic board and additional memory. The entire package cost under $6,000."

The programs to analyze the poll were written by Mr. Penn using the North Star BASIC and the computer was ready for its debut. That night, New York City experienced its second major power failure.

Since no candidate received a majority vote on Primary Day, a runoff election was scheduled a week later. "We used our personal computer to run the whole show that week and continued to use it as the major data processing element for the rest of the campaign. The day before the election, using the SOL/North Star, we called the election within a half percent of the actual vote," reported Mr. Penn. The computer was busy, even on election night, tabulating results from sample precincts. "Instant poll results are almost unheard of in our profession," said Penn, "but with our personal computer we achieved it."

Near the end of the campaign, the computer system not only analyzed the data; it helped gather it. A Mullen Opto-relay board was installed in the S-100 bus of the Sol, interfaced to eight phone lines and used to automatically dial the phones used by the poll takers.

Computer Analyzed Cows Give More Milk

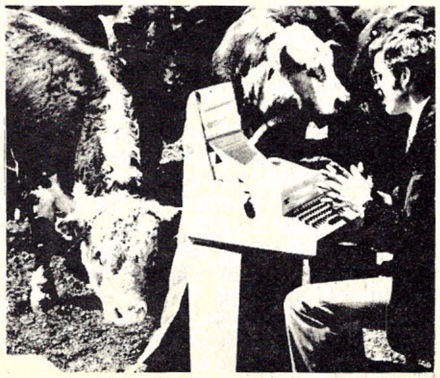
The chances are getting better that the milk you drink and the butter and cheese you eat will come from cows that are not only contented, but computer-analyzed as well.

The dairy industry today is producing twice as much milk with half as many cows as it did 15 years ago, because of computer technology.

More than 3.5 million cows nationally — about a third of the total — are being bred, fed, milked, mated and monitored for productivity with the aid of a computer. The average 'test' cow will produce 30% more milk, worth $200 more annually, than the average cow that is not being analyzed by a computer, according to Bliss Crandall, president of DHI Computing Service of Provo, Utah.

The computer produces reports for dairy farmers showing which cows are ready to be bred, which should be checked for pregnancy, which are due to calve, which should be turned dry and which should be culled from the herd. It can recommend changes in feeding according to productivity. It also offers production projections and shows an individual cow's productivity and dollar value relative to the rest of the herd.

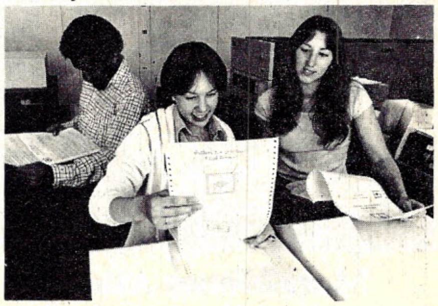
The cost to the dairy farmer for the computer analysis is about 12 cents per month per cow. But, "for every dollar a dairyman spends on computer services, he gets back $20 in increased efficiency and productivity," said Crandall.



Computer Helps Students Find Scholarship Dollars

These New Hampshire College students study possible sources of scholarship dollars found for them by a computer. The service was developed by undergraduates at New Hampshire College who programmed an IBM System/3 to match scholarship sources to an individual.

Last year, students in eight states used the computer to identify over $30 million of potential aid from local and national grant and scholarship sources. Over the next four years, scholarships available in 10 more of the nation's most populous states will be added to the computer's files. This will identify millions of additional dollars for



eligible applicants. The IBM computer currently has in its files more than 10,000 sources of aid from the six New England states, Indiana and Minnesota. Student residents in any of those states who use this service can expect the computer to check up to 1,000 potential matches and come up with an average of 20 "hits." A computer search costs $10 per applicant.

The service, called "The Fund Finder," is available through the Citizen's Scholarship Foundation of America (CSFA), a non-profit organization. Development of the data bank is sponsored by the Ford Foundation, state governments, many corporations, philanthropic organizations and private donors.

Is This A Trend?

Washington, D.C.—The Office of Education has set up a division for bright and artistically talented students. It will be placed under the Bureau for the Education of the Handicapped.

ium...compendium...co

Computer Handels Bach- Breaking Process

The Musical Heritage Society is a mail order record club (emphasizing baroque music which explains why I'm a member and thus can bring you this short story). The Society stocks several thousand different titles and puts out a monthly magazine which allows (encourages) members to order not only the current selections, but assorted back titles from stock. Records are stocked in aisles, nearly 300 feet in all..

Until recently, orders were filled (or "picked") one at a time. On average, one individual picked 30 orders per hour and got a good bit of exercise in the process — literally miles of walking. Now, however, with the aid of a computer one individual can pick 60 orders in about 10 to 15 minutes. This may seem incredible since between 200 and 400 records are involved, almost all of different numbers. What a picker now pulls from the shelves in a few minutes would require at least two hours and two or three miles of walking if a single order were pulled at a time.

Each order is ganged with 59 others upon receipt. The computer program treats these as a lot, assigning them a batch number and then numbering each order from 1 to 60. When the computer prints out the invoices for this batch, it also prints out a "picking order" sheet, ganging all the records in numerical sequence as they're stored on the shelves.

As the picker pulls records from the shelves they are placed in a "magic box", a large wheeled box so built that it has upright slots numbered 1 through 60. These slots have adjustable panels to accommodate different sized orders.

Guided by the "picking order" master sheet, the picker places the records picked in the slot numbers the computer has indicated. Thus, in a single pass down several hundred feet of shelf stock, 60 orders will have been picked.

At the end of the picking line a checker removes the orders, slot by slot, and checks the records of the numbered slot against the corresponding invoice. Upon confirmation of correctness the order is given to a packer who packs it for shipment to the post office. — *DHA*

More Money After Retirement?

"My former company gave me a retirement program that they never anticipated. By using one of their products, I'm making a lot more money now as a retiree than I ever did as a regular old working stiff."

Seventy-three year old Carrol Smith of San Jose, California, has found a new career in his retirement. Using an HP-97 programmable printing calculator, a desktop computer and years of experience in watching the stock market, he is earning more now as an investor than he ever did as a working man.

Smith began his new career with the help of an early HP desktop calculator and the original HP-35 pocket calculator. "Those early calculators were a real asset. The cost of renting a computer would have made such a small operation as mine unprofitable. Yet, the equations that I used to predict market activity required far more calculating power than I could do by hand," he said.

Smith's hobby has grown considerably in the last six years. He estimates that during that period he has moved more than $5 million in and out of the market, yielding an average income of $30,000 — all from an initial investment of about $35,000.

"Keeping track of the market requires a considerable amount of charting. Primarily, I'm searching for a number known as the 'relative strength' of a stock. This involves taking the stock price and dividing it by one of the basic market indicators — like Standard and Poor's and Dow Jones — and then using some constant multiplier to give a chartable result. The chart describes the activity of the stock in relation to the chosen indicator. From this graphic representation, a knowledgeable investor can locate 'buy' and 'sell' signals. Occasionally, the graph will give you a false signal, but 75% of the time, it will be quite accurate. As a confirming signal runs simultaneously with the graph, I use two running stock price averages, one short term and the other long. Where these average graphs cross, they confirm the relative strength signal. Other checks on relative strength are an ongoing record I keep of changes in the volume of stock sales and a 50-day true running average on the daily closing prices of each chosen stock," explained Smith.

Smith divides his computational load between the computer and the HP-97 — the large number reduction and processing operations are handled by the former; shorter, more flexible programs (like return on investment and capital gains) on the calculator. His results have been impressive, allowing he and his wife to travel on five cruises in nearly as many years and to purchase not only a mobile home, but a truck to pull it.

Computer Wasteland: U.S.A.

The largest user of automatic data processing in the World, the United States Federal Government, is also the biggest misuser, according to a two-part investigative article in *Infosystems* documenting the following facts concerning Government computer abuse:

- Automatic data processing is one of the fastest growing expense items in the National Budget. The growth of Government computers has gone from 0 in 1950 to 10,282 as of May 31, 1977, with record acquisitions currently scheduled.
- The spending has escalated so rapidly that no reliable cost figures have been developed. Estimates range all the way from $3 billion to $15 billion spent annually on ADP.
- Government computers initiate more than 1.7 billion transactions annually without evaluation, resulting in such breakdown as issuance of payroll checks to dead former employees, issuance of duplicate checks, overpayments, welfare payments to unqualified recipients, etc.
- The government computer procurement process, which can take from three to six years, "guarantees" that it will buy nearly obsolescent equipment.
- In many cases the legal requirement for competitive procurement has been virtually ignored, with supplying contractors actually writing government specifications to their own equipment.
- 47 percent of all federal dollars spent for ADP represents personnel costs, compared to 33 percent for the private sector, a huge waste in manpower and dollars.
- The civil service system, which requires the "retreading" of current personnel, rather than replacement by those technically qualified, results in incredible misuse and underuse of expensive data processing equipment.
- There is virtually no interaction or coordination between government agencies concerning data processing needs. One elected official described agencies building their respective DP inventories like a child compiling a birthday list. "They have a long list which they give to the General Services Administration and Office of Management and Budget of things they want ... and it doesn't have any relation whatsoever to what they need."
- The 1965 Public Law 89-306 (the Brooks Bill) which was passed to guarantee "the economic and efficient purchase, lease, maintenance, operation and utilization of automatic data processing equipment by Federal departments and agencies," has been "neither administered nor implemented in accordance with the intentions of Congress," according to the House Government Operations Subcommittee.

2nd Annual

PERSONAL & BUSINESS SMALL COMPUTER SHOW

NEW YORK COLISEUM
September 15-17, 1978

EXCITING

That's the word that sums up the dynamic small computer show in New York that was established last year at the Coliseum. The lecture areas were filled by interested people, the exhibits were great, the attendance was the highest of any small computer show in the country.

The big point is that everyone who came went away happy. Accountants, hobbyists, lawyers, doctors, brokers, retailers, business people, programmers, research scientists and just plain homeowners.

The second big point is that we're doing it again. This time, bigger and better. Top seminars, top exhibits. Small computers from $500 up, software and kits.

Save time. Get your tickets in advance by mail.

NAME ______________________________
(please print)

ADDRESS ______________________________

City ____________ State ________ Zip ________

Send me tickets in advance for Personal & Business Small Computer Show, Sept. 15-17, 1978, in the New York Coliseum.

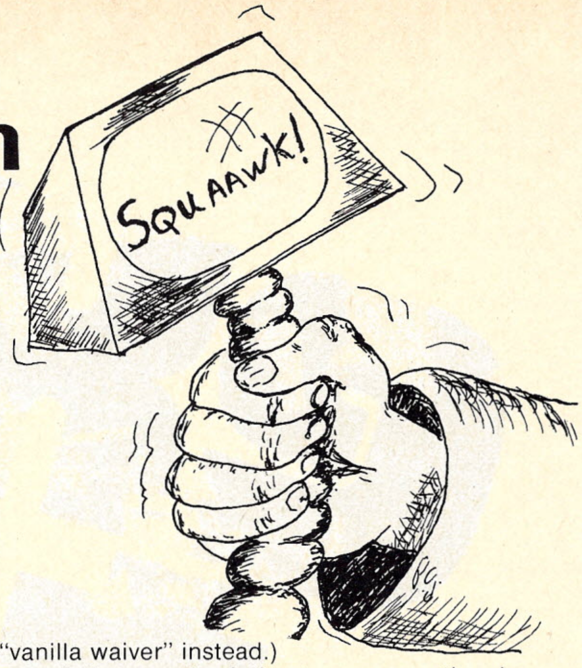
(Check One) One Day ($5)__ Two Days ($9)__ Three Days ($13)__

Send check or money order (U.S dollars) payable to Personal & Business Small Computer Show to 78 E. 56th St., New York, N.Y. 10022.

PERSONAL & BUSINESS SMALL COMPUTER SHOW
78 East 56th Street, New York, N.Y. 10022

CIRCLE 165 ON READER SERVICE CARD

New 8K CAI System Can be Molded to Fit Users' Needs



Michael G. Shafto
Peter B. Worland

During the past decade several large systems have been developed to support Computer Assisted Instruction (CAI). These systems are characterized by the use of sophisticated graphics and modern question-answering programs.

Although CAI on these systems is achieving some success in educational settings, and is even expanding into business/industrial markets, the currently available systems are limited in the following ways: First, hardware costs prohibit ownership except by the largest universities and corporations. Second, line charges and equipment rental costs are quite burdensome for small users unless they are very near the host institution. Finally, CAI software is almost always machine-dependent and non-portable.

Dumb User Heuristics (DUH), Inc., has recently placed on the market a remarkable and revolutionary CAI system that transcends all three of these limitations. This elegant system, called PLA-DOH, is written in the flexible macro-language TOOTLE. It will run on any computer or hand-held calculator with 8K user memory space and a TOOTLE interpreter. Portability is guaranteed since TOOTLE is written in the very popular ALGOL 58, and will thus compile on any machine with an ALGOL 58 compiler.

TOOTLE is a simple language to use and includes numerous control structures to facilitate the structured design of new user applications. Structures included are: DO-WHILE, DO-UNTIL, NEVER-DO, WHY-DO-WHILE?, WHY-NOT-GO-TO?, the very popular GO TO-COME FROM combination, and the powerful IF DO NOT SUCCEED TRY AGAIN command.

Although an 8K memory is sufficient for the complete DUH PLA-DOH system, response time can be reduced by 95 percent if a 512K increment is added. On a typical 8K machine, the mean response time (using a first-grade addition curriculum) is 20 minutes. As standard equipment, however, a sketchpad, a box of jacks, and a book of crossword puzzles are provided with each PLA-DOH terminal.

DUH provides a number of flexible peripheral devices to enhance the PLA-DOH system. These peripherals are manufactured by DST (Destructible Systems Technology), noted pioneers in the recycled transistor industry. The standard CRT features a bulletproof, non-glare screen with purple and chartreuse display. Up to 4000 characters per line can be displayed (but at least 500 characters *must* be displayed). There is no conclusive proof that looking at this display for 15 minutes can cause permanent eye damage, but DUH does require each user to sign a "nominal waiver." (At the user's option, he/she may sign a "vanilla waiver" instead.)

DUH is especially proud of the two-way optional SQUAAWK feature which enhances the interactive aspect of the CAI experience. Often the CAI user may become frustrated by a segment of insensitive dialogue; for example,

PLEASE RETYPE USER NUMBER
USER NUMBER DOES NOT EXIST
USER DOES NOT EXIST
. . . no response . . . etc., etc.

At this point, the PLA-DOH user may grasp the terminal around its rubber neck and squeeze and/or shake it at any desired pressure and/or frequency. The CRT will emit a bloodcurdling "SQUAAWK!", and, if the abuse is severe enough, will turn blue.

The inverse SQUAAWK feature provides suitably-timed negative reinforcement or punishment to the user. The squeeze pressure and frequency of shaking of the user by the CRT are controlled by parameters set by the user's institution.

For users unable to afford the innovative SQUAAWK feature, PLA-DOH also provides more conventional Verbal Abuse, guiding the student's progress with motivational expletives; for example,

WHERE DID YOU LEARN TO SPELL — IN THE MEN'S ROOM?
WHO TURNED ON THE TERMINAL FOR YOU?
YOUR TIME IS UP. PLEASE RETURN TO THE WARD.

DUH, Inc., has developed a number of software packages to help prospective users adapt the PLA-DOH system to their specific needs. These include

(a) "Computer Crime — 10 easy ways to make your fortune,"

(b) "Challenging Chess," in which the computer cleverly eliminates opposing pieces from the board and/or restores its own pieces after they are captured. Alert users may also cheat if they can figure out the "rules" the computer is using.

(c) "Doodles from TOOTLE," an entertaining and educational graffitics package.

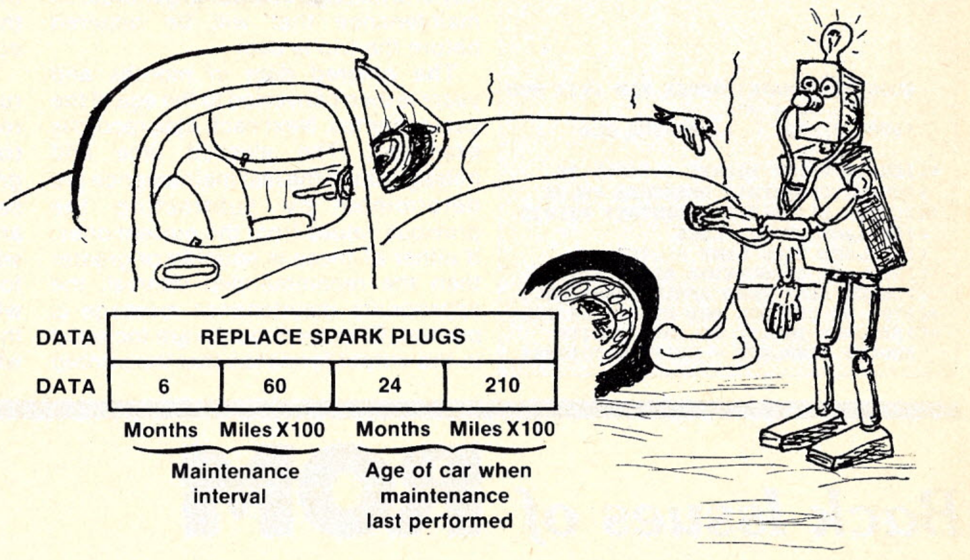
(d) "The Joy of Sex," a *really* graphic package (requires floating topless software).

In summary, DUH has developed a system which is more flexible and less expensive (that is, cheaper) than any other on the market today. As the PLA-DOH system is shaped and molded to fit more and more users' needs, DUH looks forward to overtaking the giants of the CAI industry. As our motto says, "I think I can . . . I think I can . . . I think I can." ■

Does Your Car Need Oil?

Rob Lufkin

Anyone who has ever owned an automobile or similar type of complicated equipment has probably, at one time or another, been faced with the problem of determining when maintenance is due. It's not difficult to conclude that emergency brake repair is necessary if pushing the brake pedal fails to slow the car. However, when the maintenance is routine and the indications are that checkup A is to be done every two months or 1200 miles, replacement B every six months or 3000 miles, and inspection C every 24 months or 10,000 miles, the problem is more complex. Now it is possible, by entering the correct date and mileage into a microcomputer, to be informed of all routine maintenance that is required.



Data record that says "Replace spark plugs" every 6 months or 6,000 miles. This was last done when the car was 24 months old with a mileage of 21,000

Program

The program was written with the small-system user in mind. It uses less than 4K of memory and contains all necessary information in data statements so that no file manipulations are needed. The database in the example is for an automobile, although it may be easily modified for other applications.

Database

The database consists of one data record for each maintenance reminder. The data record is contained in a pair of data statements. The first statement contains a message for that record such as "Replace air pump air filters." The second statement contains four time and mileage variables. The first two are the intervals in months and miles at which the maintenance is to be performed. To simplify storage, the time units are all converted to months and mileage is expressed in hundreds of miles. The last two variables are the time and mileage when the work was last performed. Here the time is expressed as the age of the car in months (starting from the time the car was first driven). If, for example, the maintenance has never been performed or the car is brand-new, the last two time and mileage variables are both set to zero.

Program Operation

The program works by asking for the mileage of the car and the date. These values may be either for the current

```
0010 REM MAINTENANCE PROTOCOL GENERATOR
0020 REM BY ROBERT LUFKIN
0100  M1=8: Y1=77: REM  CAR'S BIRTHDAY
0200 INPUT "MILEAGE (X100) ",M5
0300 INPUT "MONTH, YEAR  (EX: 2,78) ",M2,Y2
0400  Y3=Y2-Y1: M3=M2-M1: T3=M3+(12*Y3)
0425 PRINT CHR$(147): PRINT: PRINT: PRINT
0430 PRINT "  SUGGESTED MAINTENANCE FOR 1977 MGB"
0440 PRINT "             ON ";M2;"/";Y2
0450 PRINT "  ";STR$(M5);"00 MILES     ";T3;" MONTHS OLD"
0700 PRINT : FOR I=1 TO 13
0710 READ X$, T4, M4, T6,M6
0715  M7=M5-M6: T7=T3-T6
0720 IF M7>=M4 THEN 800
0730 IF T7>=T4 THEN 800
0740 GOTO 850
0800 IF M7>=1.5*M4 THEN 809
0802 IF T7>=1.5*T4 THEN 809
0805 PRINT " []";: GOTO 810
0809 PRINT "*[]";
0810 PRINT X$
0850 NEXT I
0855 PRINT
0860 PRINT "  REVIEW COMPLETE - HAPPY MOTORING!"
0900 DATA "PERFORM OSCILLOSCOPE TUNE-UP",12,120,5,120
0903 DATA "CHECK VALVE ROCKER CLEARANCES",6,60,0,0
0906 DATA "CHANGE OIL FILLER CAP",12,140,5,120
0909 DATA "CHECK CRANKCASE BREATHER VALVE",12,120,0,0
0912 DATA "CHECK DISTRIBUTOR CONTACT POINTS",6,60,0,0
0915 DATA "REPLACE SPARK PLUGS",12,120,0,0
0918 DATA "REPLACE FUEL LINE FILTER",12,120,0,0
0921 DATA "REPLACE CARBURETOR AIR CLEANERS",12,120,0,0
0925 DATA "CLEAN OVERDRIVE FILTERS",24,240,0,0
0928 DATA "CLEAN BRAKE LININGS, DRUMS, AND DISKS",12,120,6,
0931 DATA "CHECK REAR ROAD SPRING SEAT BOLTS",12,120,0,0 130
0934 DATA "TOP UP GEARBOX AND REAR AXLE OIL",6,60,3,30
0937 DATA "REPLACE ENGINE OIL AND FILTER",6,60,2,10
1000 END
```


Rob Lufkin, Star Rt. 1, Box 4, Charlottesville, VA 22901.

OIL continued...

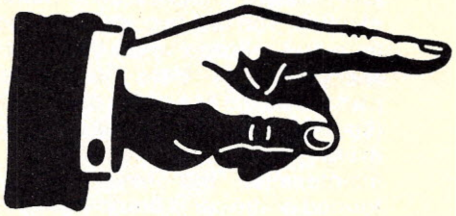
```
MSI READY
#RUN
MILEAGE (X100) ? 150
MONTH, YEAR  (EX: 2,78) ? 1,78


  SUGGESTED MAINTENANCE FOR 1977 MGB
         ON 1 /78
  15000 MILES      5  MONTHS OLD

*[]CHECK VALVE ROCKER CLEARANCES
 []CHECK CRANKCASE BREATHER VALVE
*[]CHECK DISTRIBUTOR CONTACT POINTS
 []REPLACE SPARK PLUGS
 []REPLACE FUEL LINE FILTER
 []REPLACE CARBURETOR AIR CLEANERS
 []CHECK REAR ROAD SPRING SEAT BOLT
*[]TOP UP GEARBOX AND REAR AXLE OIL
*[]REPLACE ENGINE OIL AND FILTER
  REVIEW COMPLETE - HAPPY MOTORING!
```

day, or, if a long journey is to be taken, adjusted values may be entered to generate future maintenance that will be required during the course of the trip. For example, if the trip is to cover 2,000 miles and be two months long, these amounts are added to the current day and mileage in order to generate all maintenance that will be required before the trip is over.

The entered date in months and years is then converted to the age of the car in months. Next each data record is examined. The elapsed time and mileage since the last maintenance are determined by subtracting the previous values from the current ones. If either of these is equal to or greater than the recommended interval, the appropriate maintenance message is printed. Check-off boxes are included in the output format so that the printed output could be easily used by an automobile dealer or repair shop as a work list. Work that is overdue by 50% or more of the maintenance interval (either mileage or time) is flagged with double asterisks.

After the maintenance is performed, the data statement is merely updated to the age of the car when the work was completed.

The program has been tested and run in its present form with good results. In addition to providing distraction for the computerist who has grown weary of hunting Klingons, it serves simple but useful functions around the home. Not the least of which is to provide yet another answer for the computerist's amazed friends who invariably ask, upon first seeing the electro-cybernetic wonder, "But what's it good for?" ■



READERS!

ROM Magazine is now contained in *Creative Computing.* We'll be running 32 pages or so of ROM material including columns from their (now our) very talented and creative contributing editors. For more information, see page 4.

JUDO AND THE BLACK-BELTED COMPUTER

AI, THE RESTAURANT GAME, AND TOP-DOWN HOLISM

HOW TO MAKE YOUR COMPUTER FEEL AT HOME

MICRO MATRICES: Getting the Math off Your Back

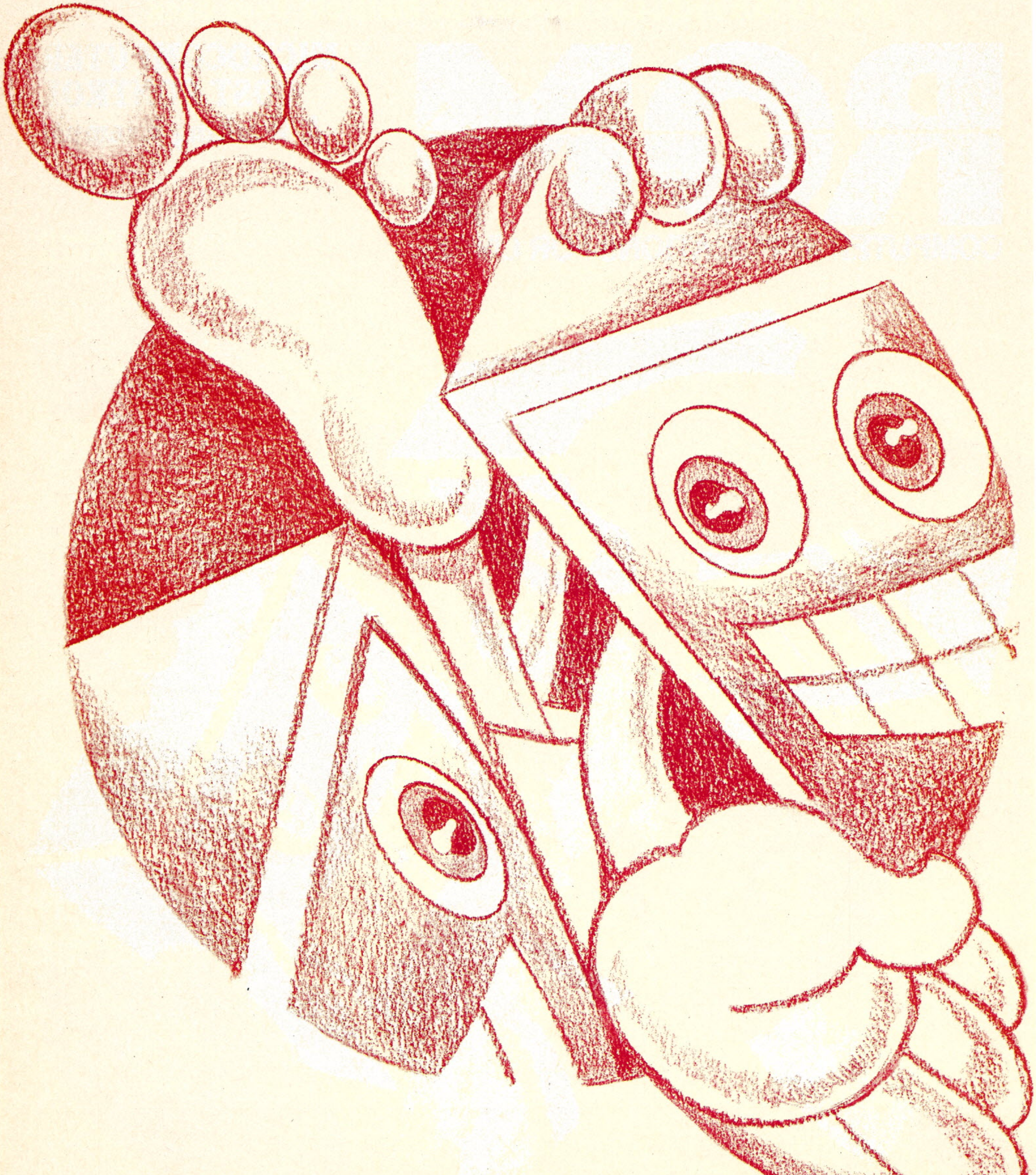
ROM

COMPUTER APPLICATIONS FOR LIVING

MICROCOMPUTER COST CONTROL
A Working Example

Volume I, Number 10
May 1978/$2.00

JUDO



&

The Black-Belted Computer

by Ramon C. Barquin

The use of computers in sports applications has been quite steadily increasing over the last few years, and the impact on some sports has been dramatic. Some years back, a fall joint computer conference acknowledged the field's growing importance by dedicating a session to computers in sports. One of the applications least publicized in this area has been the scoring and general record keeping of international judo competitions. Utilized for the first time in the VI Pan American Judo Championship held in 1968 in San Juan, Puerto Rico, the JUDO system met with such wide acceptance that it was adopted by the International Judo Federation (IJF) for further use in world-class events. A revised version of the system was then utilized by the IJF during the 1969 VI World Championship in Mexico City and has been used in every world championship ever since and in the Olympics since 1972.

Judo, the "gentle way"—initially developed by Dr. Jigoro Kano—is a relatively new addition to the international family of sports, though its roots actually date back more than a millenium. Dr. Kano was born near Kobe, Japan in 1860. After a number of years studying the forms of jiu-jitsu, or empty-handed fighting, he established the Kodokan School, where he taught his own system which he called judo, or the way of gentleness. The Kodokan, which he founded in 1882, is presently headed by his son, Dr. Risei Kano.

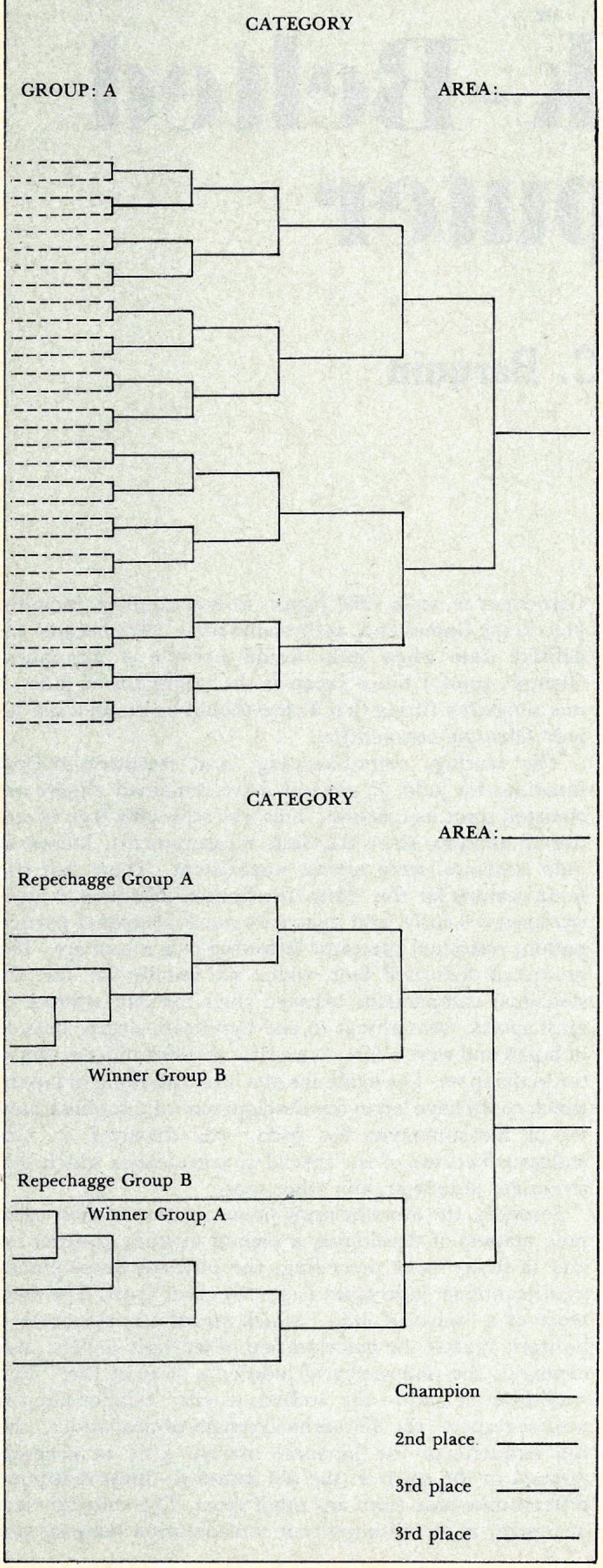
Since 1953, the International Judo Federation has been in existence and has periodically sponsored world championships. The IJF, after a few invitational tournaments, regimented the championships to be a biennial event. But it was only in 1964, though, at the Tokyo Olympic Games, that the sport was incorporated into Olympic competition. (When a new sport is voted into the Olympic Games, a period of eight years is given for each country to prepare its own teams and delegations in the sport. Every host country may include a sport which is not part of the Olympic Games per se, so in 1964 Japan, as host country, included judo in the Games and, at the same time, 1972 became the definite date when judo would become a permanent Olympic sport.) Since Japan is the originator of judo, it was altogether fitting that Tokyo should be the first site for judo Olympic competition.

The scoring, record-keeping, and statistical-analysis functions for judo, in general, have remained almost unchanged since its creation. This was especially true of statistical analysis, since the basic measurements, known as judo statistics, were almost nonexistent. There are two main reasons for this. First, the limited diffusion of judo, until very recently, and its strictly nonprofessional participation, restricted spectator following to a minimum. The groups of dedicated fans, which are usually the base for statistical comparisons between their favorite athletes in most sports, were absent to any significant degree outside of Japan and were a direct cause for the lack of basic statistics in the sport. The existence of a loyal following in Japan, which could have led to the development of a sophisticated set of measurements for judo, was thwarted in this endeavor because of the special characteristics which differentiate judo from any other sport.

Secondly, the overwhelming preoccupation of most early judo masters of developing a mental attitude parallel to, and in many cases preceding, the physical development significantly set judo apart from any other sport. The existence of a "spirit of judo," which steadfastly stands as a bulwark against the deterioration of its basic beliefs, also expounds the philosophy of judo as a "way of life." The traditions of judo—the student-teacher relationship of utmost respect; the hierarchical system of gradations; the full etiquette of the Japanese martial arts, in general, imbued in the spirit of the old samurai—most definitely differentiate judo from any other sport. The classical view of combat as the all-important confrontation between two

Illustrated by Rex Ruden

Figure 1
Elimination Process



mighty opponents precluded much attention being given to anything but the action itself. In the wake of the ancient duels between samurai, which most often were settled by death, record keeping and statistics seemed totally out of place.

After World War II, with the diffusion and popularization of judo the world over, international competition and non-Japanese participation slowly changed the overview and scope of the martial art to allow its acceptance as a sport proper. With competition on an international scale came the rivalries which give rise to the need for comparisons. The stage was set for the appearance of formal measures with which to carry out these comparisons between judokas, or between "dojos" (a dojo is the club, or locale, where the practice of judo takes place), or between countries in their competitions against each other.

Obvious as the need was for developing this basic set of judo statistics, not very much was done during the 1950s or early 1960s. Most of the advances during that time were dedicated to devising good, applicable score-keeping and elimination systems. Thus, the round-robin, double-elimination, single-elimination systems gave way to the more sophisticated methods developed to handle the ever increasing number of participants and spectators. The two methods most utilized internationally today are the "repechagge," introduced by Brazil and adopted by the IJF in the early 1960s, and the "five bad point system," devised by Rick Merten of the United States Judo Federation (USJF).

Records of world championships, of course, were kept by the IJF, but with very little information beyond the names and countries of medalists. The Japanese kept some records of competition in a purely descriptive manner, relating the action as it occurred much in the manner of a sportscaster during a boxing match or a baseball game. Results of combats were often recorded with some additional information, but with no apparent intention of later analysis, since they were soon destroyed. The interest with which the computerized statistics and results of the 1968 Pan American Championship were greeted by IJF officials and some old Kodokan instructors gives credence to the fact that very little was done in the way of statistical analysis with any records which might have been kept in Japan until that time. In truth, the only such study appearing in judo literature has been some statistical tabulations from the All-Japan High School Contests presented in *Judo for High School,* a publication of the USJF's Interscholastic Committee, in 1970. (At least this has been the only such work I found or which has been mentioned to me by IJF officials.)

One of the basic difficulties in doing any type of analysis in the sport stemmed from the fact that the fundamental measures had not been defined. For example, in baseball, batters are measured by their batting averages, runs-batted-in, and the number of home runs they hit, while pitchers are compared by their won-lost averages, and their earned-run averages (ERA). And for that matter, the percentage of completions, yards-gained-rushing, yards-gained-passing, and time of possession are today essential to the players, spectators, and coaches of an American football team. In the sport of judo, the equivalent measures were not properly defined. Enter the computer. Used for the first time in 1968, the cornerstone was set in developing this hard core of statistics for the sport. Almost as a byproduct, in an attempt to automate scoring and record keeping, the systems engineering exercise yielded the basic set of mea-

sures from which constructing a statistical foundation for judo, as a sport, could proceed.

On realizing the first steps in the task of computerizing the scoring and record keeping of a judo tournament, it became evident that the parameters and measures would have to be defined from scratch. Almost intuitively, from practical competitive experience, a feeling develops for the factors which should be highlighted and measured within judo competition. Logically, these factors will have to deal with combats won, combats lost, and the ratio between them (a won-lost average). Also, they must deal with a breakdown, or specific accumulations, by the manner in which the fight was won or lost. The elapsed time in each combat, the cumulative time of competition by judoka, the average time—all will be of equal interest. "Ippon," or one point, being the ideal form of victory, it might be worthwhile to keep the percentage of these to overall combats, or victories, for each team, country, or individual contestant. In addition, it would be valuable to determine and analyze other things, for instance, frequency of utilization of the different techniques scored by or against the specific countries, and the absolute tallies of these techniques which could be used for theoretical analysis. Or, in order to know what to expect from the different delegations, their vital and noncompetitive statistics such as weight, height, rank, and years of experience may also be needed.

From these factors, then, an almost natural division of judo statistics can be derived. Statistics can be drawn from either static or from dynamic data, depending on whether the information stems from the actual combat or not. In the first group go all those measures which come from noncombat sources, and in the second, or dynamic group, those which are derived from combat information. For example, the average weight or height of a delegation, their cumulative years of experience, or an individual contestant's rank are all typical of static information. Thus, these will constitute, or become input to, static judo statistics. On the other hand, the life won-lost average of a competitor or of a team during a specific contest, or the percentage of

Figure 2
Log of Combats

WORLD JUDO CHAMPIONSHIP
LUDWIGSHAFEN, B. P. D.
08/28/71—09/06/71

CATEGORY UNDER 63K — DATE 9/04/71

LOG OF MATCHES

1	DAVID GIVNEY	OF AUSTRALIA	DEFEATED MOISTRI CHOMSAKORN	OF THAILAND
	BY IPPON	IN 5 MIN	WITH KAMI-SHIHO-GATAME	
2	J. J. MOUNIER	OF FRANCE	DEFEATED J. BONICH	OF SPAIN
	BY IPPON	IN 4 MIN 59 SECS	WITH MUNE-GATAME	
3	FERENC SZABO	OF HUNGARY	DEFEATED T. CENICEROS	OF MEXICO
	BY YUSEI-GACHI	IN 6 MIN		
5	ANELSON GUERRA	OF BRAZIL	DEFEATED WILLY VARGAS	OF PUERTO RICO
	BY YUSEI-GACHI	IN 6 MIN	WITH KATA-GURUMA	FOR WAZARE
6	CHOI JONG SAM	OF KOREA	DEFEATED A. NIANG	OF SENEGAL
	BY IPPON	IN 32 SECS	WITH IPPON-SEOI-NAGE	
7	FRANZ FISCHER	OF GERMAN FED REP	DEFEATED I. HAMBALEK	OF SWEDEN
	BY IPPON AND	IN 2 MIN 18 SECS	WITH KESA-GATAME	FOR WAZARE
	HARAI-GOSHI		FOR WAZARE	
8	G. VISMARA	OF ITALY	DEFEATED RODNEY PARR	OF UNITED STATES
	BY YUSEI-GACHI	IN 6 MIN	WITH KATA-GURUMA	FOR WAZARE
9	GUSTAAF LAUWEREINS	OF BELGIUM	DEFEATED AN. FREDERICI	OF NETH. ANTILLES
	BY IPPON	IN 28 SECS	WITH HARAI-GOSHI	
10	PIZCHELAURI SCH.	OF U.S.S.R.	DEFEATED DIETER SCHOLZ	OF GERMAN DEM REP
	BY YUSEI-GACHI	IN 6 MIN		
11	P. WILER	OF SWITZERLAND	DEFEATED K. H. WERNER	OF GERMAN DEM REP
	BY YUSEI-GACHI	IN 6 MIN		
12	JANOS PULAI	OF HUNGARY	DEFEATED KIM SANG CHUL	OF KOREA
	BY YUSEI-GACHI	IN 6 MIN		
13	MITCHELL KAWASAKI	OF CANADA	DEFEATED LUIS SHINOHARA	OF BRAZIL
	BY YUSEI-GACHI	IN 6 MIN		
14	ISWANDA. SETIWAH	OF INDONESIA	DEFEATED ROBERTO PACHECO	OF PUERTO RICO
	BY IPPON	IN 1 MIN 20 SECS	WITH UCHI-MATA	

ippons, or cumulative time of combat for a delegation are typical of dynamic judo statistics.

In the actual implementation of JUDO, a small subset of parameters was selected for computation and presentation in order to fulfill the general goals of the system without overtaxing the overall costs. An overview of these measures is presented in outline form:

Static

A. Number of judokas (2,3,4)
B. Number of dojos (2)
C. Years of judo practice (1,2)
D. Size delegation (2)
E. Number of Black Belts (2)
F. High rank (2)
G. Low rank (2)
H. Avg. weight (2,4)
I. Avg. height (2,4)
J. Avg. age (2,4)
K. Avg. rank (2,4)
L. Avg. experience (2,4)
M. Height (1)
N. Weight (1)
O. Age (1)
P. Rank (1)
Q. Experience (1)
R. Coach rank (2)
S. Coach experience (2)

Dynamic

A. Won (1,2)
B. Lost (1,2)
C. Won-lost avg. (1,2)
D. Avg. time per match (1,2,3,4)
E. Cum. time of match (1,2,3,4)
F. Pct. won by ippon (1,2,3,4)
G. Performance points (1,2)
H. Medals won (1,2)
I. Won by ippon (1,2,3)
J. Won by yuseigachi-wazare (1,2,3)
K. Won by other (1,2,3)
L. Won by yuseigachi-no wazare (1,2,3)
M. Lost by ippon (1,2,3)
N. Lost by yuseigachi-wazare (1,2,3)
O. Lost by yuseigachi-no wazare (1,2,3)
P. Lost by other (1,2,3)
Q. Won against each country (2)
R. Lost against each country (2)
S. Ippons with each technique (2,3,4)
T. Wazares with each technique (2,3,4)
U. Pct. totals of S, T (2,3,4)
V. Yukos scored
W. Yukos scored against
X. Kokas scored
Y. Kokas scored against
Z. Penalties

Note: The numbers in parentheses indicate statistics gathered by:

(1) Individual competitor
(2) Country
(3) Weight division
(4) Complete tournament, overall

There are three basic sources for all these statistics: the individual competitor, the country, and the combat itself. In fact, the following is a breakdown of source information:

Competitor

Name
Height
Weight
Age
Rank
Experience
Country represented
Weight division
ID number

Country

Name
Size delegation
Number of dojos
Number of Black Belts

JUDOKA	WON	LOST	AVG.	AVG. TIME PER FIGHT MIN.
UNITED STATES				
LARRY FUKUHARA	1	1	.500	1.97
ISAO MURA	1	2	.333	2.97
PAUL K. MARUYAMA		1	.000	6.00
TOSHIUKI SEINO		2	.000	3.51
IRWIN L. COHEN		2	.000	3.17
CLYDE WORTHEN	2	1	.667	3.17
RODNEY R. HAAS		1	.000	6.00
DOUGLAS GRAHAM	1	1	.500	5.18
ALLEN COAGE	1	1	.500	5.17
WILLIAM D. MCCAULEY		1	.000	3.00
DALE LEHMAN	1	1	.500	3.50
RICHARD R. WALTERS	2	1	.667	4.47
********	9	15	.375	3.83
			TOTAL	MATCHES 24
U.S.S.R.				
SERGEI SUSLIN	4	2	.667	3.44
DAVID RUDMAN	4	2	.667	3.58
A. TSIUPACHENKO	1	1	.500	3.87
VLADIMIR POKATAEV	4	1	.800	5.42
KIBROTSASHVILI	2	2	.500	3.38
GIVI ONASHVILI	3	1	.750	4.53
A. KIBROTSASHVILI		2	.000	.00
GIVI ONASHVILI	1	1	.500	6.00
********	19	12	.613	3.89
			TOTAL	MATCHES 31

MEDALS

Years of judo
Population
Coach name
Coach rank
Coach experience
ID number

Combat

Number of combat
Winner ID
Loser ID
Time of combat
Decision type
Techniques used by winner
Techniques used by loser
Number of yukos
Number of kokas
Number and type of penalties

There are various ways in which a judo match can be won. These are: (1) ippon, or one point; (2) yuseigachi, or decision; (3) hansokumake or sogogachi, which are penalty and combination half-point penalty decisions; or (4) fusensho, or default. An ippon can be achieved directly or by two wazares, or half-points. The decision, when there is a wazare already scored, should be differentiated from the one when there is not. Yukos and kokas are two levels of advantages scored below wazare which may decide a match. Penalties of different levels of seriousness are converted to these scores at the end of a match. (For a detailed description of the ways to win a judo match, see the *Handbook of the International Judo Federation,* London, 1976.)

Information on techniques does not necessarily have to exist, since a match may be won by decision and not by technique used. On victories by default or penalties the same holds true. For a winner by ippon, one or two movements may appear, depending on whether he has scored for ippon after having scored a wazare, or whether he scored ippon through two wazares, or whether he scored ippon without a prior wazare. For a winner by yuseigachi-wazare only the technique with which he scored the wazare will appear. For a loser, the only possibility for a technique

Figure 3
Breakdown of Combats by Judoka

TOTAL TIME OF COMBATS	CAT.	WEIGHTED AVG.	VICTORIES WON BY IPPON	WON BY Y-GACHI	WON BY OTHER	DEFEATS LOST BY IPPON	LOST BY Y-GACHI	LOST BY OTHER	MEDAL
3.94	1	.500	1			1			
8.92	1	.333	1			2			
6.00	2	.000					1		
7.02	2	.000				1	1		
6.34	3	.000				2			
9.51	3	.667	2			1			
6.00	4	.000					1		
10.36	4	.375		1		1			
10.34	5	.500	1				1		
3.00	5	.000				1			
7.00	6	.250			1	1			
13.40	6	.667	2				1		
91.83			7	1	1	10	5		
		WON BY IPPON 29.17%							
20.63	1	.583	2	2		2			BRONZE
21.48	2	.667	4			1	1		BRONZE
7.74	3	.500	1				1		
27.11	4	.700	2	2			1		BRONZE
13.53	5	.500	2			2			
18.12	5	.688	2	1			1		BRONZE
.00	6	.000						2	
12.00	6	.375		1			1		
120.61			13	6		5	5	2	
		WON BY IPPON 41.94%							
			BRONZE 4						

to appear is if he had scored wazare on his opponent prior to losing by ippon, or by having the decision go against him at the end of the match. In a sogogachi decision, which is a penalty combined with a wazare, the technique will also appear in the winner's or loser's position. Techniques scoring yukos or kokas have not been generally recorded.

Some additional information has been captured occasionally to produce special reports. For example, associating each match with the officials who judged to give a feeling for the quality of refereeing upon analysis, and differentiating right- and left-sided techniques for a more penetrating study into the effectiveness of throws.

The JUDO system is composed of approximately twenty computer programs designed to maintain running scores under the repechagge or five bad point system and, at the same time, store all pertinent information concerning the combat and its participants, providing various statistical reports from said data. There are five basic modules into which the programs can be subdivided. These are:

I. Initial random-pairing program
II. On-line scoring program
III. On-line referee-selection program
IV. Report-producing programs
V. Maintenance programs

These areas will be treated in more detail later on, as the different outputs from the various sectors are described and analyzed.

Originally designed and implemented to run on an IBM 360/30 with DOS (disk operating system) and 64K of memory, the package is extremely flexible and has been adapted to run on a number of other compatible machines and operating systems. Because of the necessity of conforming to the configurations available to the IJF at a specific site of competition, JUDO has been run in batch mode on various IBM S/360s and S/370s with DOS or OS (operating system) and on an IBM 360/20. Minimal configuration includes at least 32K of memory, a card reader, a printer, and at least one disk drive for file storage. The most recent tournaments—the 1975 World Championship in Vienna and the 1976 Olympic Games in Montreal—have been run on-line with TSO on large IBM 370s.

Due to the initial circumstances and availability of software in the development of the system, the component programs were written using more than one programming language. The nonuniformity of the package, as far as the source languages are concerned, thus denote a definite tendency toward fast implementation, sometimes detrimental to more efficient operation and maintenance. The infrequency of the use of JUDO, however, with world championships taking place only once every two years and Olympic games every four, did not place, initially, a high premium on operations and maintenance efficiency. Of course, as utilization of the system continues to increase, and as national federations start to adopt and implement it, then the necessary steps should be taken to standardize the programming. At the present stage, JUDO features coding in four programming languages. The distribution within the five basic sectors, with approximate percentages by sector, are as follows:

I. Initial random-pairing program (PL/I), 10%
II. On-line scoring program (Assembler), 15%
III. On-line referee-selection program (PL/I), 10%
IV. Report-producing programs (RPG, FORTRAN), 50%
V. Maintenance programs (RPG, Assembler), 15%

Depending on the software limitations of a specific system, FORTRAN could easily be converted to PL/I, as could the Assembler programs. The RPG-coded programs, which constitute the majority, would best remain as such due to the adequacy of the language for easily producing the reports with which the system is so directly concerned.

The general operation of JUDO is subdivided into various phases which utilize programs from one or more of

Figure 4
Summary of Techniques Scored With and Against

	TECHNIQUE	TORI	UKE
GREAT BRITAIN	Uchi-mata	2	
	Kouchi-gari		1
	Harai-makikomi		1
	Kesa-gatame		2
	Kuzure-kami-shiho-gatame	1	1
	Kuzure-yoko-shiho-gatame		1
ITALY	Tai-otoshi	1	
	Seoi-nage		1
	Uchi-mata		1
	Uchi-mata-skashi		1
IVORY COAST	Uchi-mata		1
	Ouchi-gari		1
JAPAN	Tai-otoshi	6	3
	Seoi-nage	4	
	Ippon-seoi-nage	3	
	Seoi-otoshi	1	
	Sukui-nage	1	1
	Harai-goshi	1	
	Tsuri-komi-goshi	2	
	Uchi-mata	12	
	Kaeshi-waza	1	
	Ouchi-gari	4	
	Osoto-gari	6	
	Sasae-tsuri-komi-ashi	1	
	Kouchi-gari	3	
	Kosoto-gari	1	1
	Soto-gake	3	
	Tomoe-nage	3	
	Sumi-gaeshi	2	
	Harai-makikomi	1	
	Uchi-mata-skashi	1	
	Kesa-gatame	1	1
	Kuzure-kesa-gatame	1	1
	Kuzure-kami-shiho-gatame	6	
	Yoko-shiho-gatame	1	

the five basic modules. These operating phases can be outlined as follows:

A. File creation and edit
B. Initial random-pairing generation
C. Vital statistics and participation report generation
D. Competition scoring
E. Referee selection
F. Log of combats generation
G. Generation of final reports

As can be deduced from the scheme, phases A, B, and C are one-time phases executed prior to the competition. Phases D, E, and F are cyclic and are done once for each weight class. Phase G is also one-time and carried out after the tournament has ended.

PHASE A
File Creation and Edit

Initial files and tables are created after editing runs insure correctness of the data. The tables and files are grouped into different sets.

Tables

1. Competitors
2. Countries
3. Judo techniques

Files

1. Country
2. Competitors
   a. By country
   b. By weight division
3. Combats won or lost
   a. By country
   b. By weight division
   c. By techniques utilized

PHASE B
Initial Random-Pairing Generation

When the exact number of participants for each weight class are known, the initial pairing is produced by randomly selecting judokas to an A or B pool. Then they are placed in a position within the pool. The pools are allowed only one competitor per country, and their size and "bye" positioning procedure is predetermined by the rules of the repechagge method. If the five bad point system is used, then no pools are necessary, and only random selection of slots for the competitors within a list for each division is done. If the number of competitors is odd, the last man on the list draws a bye the first time around.

PHASE C
Vital Statistics and Participation Report Generation

During this phase, two reports are generated: those dealing with the vital statistics and personal information of the competitors, and those dealing with the general data and delegation averages of the participating countries. For the first type of reports, lists are produced by country, containing information such as the name, height, weight, age, rank of each competitor and the head coach as well as a computation of each delegation's average height, weight, age, and rank. The general tournament report gives, in addition to these delegation averages, such general information by country as population, number of judokas, number of dojos, number of Black Belts, and the years since judo was introduced in the country. It also tallies these for final totals and produces a "typical tournament competitor" with vital characteristics equal to the average of all competitors.

PHASE D
Competition Scoring

Upon completion of each combat, certain information is captured and either entered directly onto the system

Figure 5
Waza Technique Utilization

WAZA	TECHNIQUE	FREQUENCY	
Te-waza	Seoi-nage	17	
	Tai-otoshi	16	
	Ippon-seoi-nage	4	
	Sukui-nage	4	
	Seoi-otoshi	2	
	Morote-gari	2	
	Morote-nage	1	**46
Koshi-waza	Uchi-mata#	26	
	Harai-goshi	14	
	Tsuri-komi-goshi	3	
	Kaeshi-waza	3	
	Sode-tsuri-komi-goshi	2	
	Koshi-waza	1	**49
Ashi-waza	Ouchi-gari	15	
	Osoto-gari	10	
	Kouchi-gari	7	
	Sasae-tsuri-komi-ashi	5	
	Kosoto-gari	4	
	Kosoto-gake	3	
	Soto-gake	3	
	Ashi-guruma	2	
	De-ashi-harai	1	**50
Ma-sutemi-waza	Tomoe-nage	4	
	Sumi-gaeshi	2	
	Ura-nage	1	** 7
Yoko-sutemi-waza	Tani-otoshi	1	** 1
Makikomi-waza	Harai-makikomi	3	
	Soto-makikomi	1	** 4
Counters	Uchi-mata-skashi	2	
	Kosoto-skashi	1	** 3

#Uchi-mata is sometimes considered a hip technique and sometimes a leg technique. For our purposes we have classified it as a hip technique.

through the on-line terminal or coded for future keypunching. The data consists of: winner number, loser number, time of combat, type of decision, techniques used by winner, techniques used by loser, category of weight, the sequential number of the combat, the number of yukos and kokas, penalties, and identification of judges and referees. Simultaneously, losers are tagged and eliminated according to repechagge rules kept on a chart similar to the one presented in figure 1. This chart permits a cross-checking procedure with the listing of records produced from all the matches. The actual scoring takes place, of course, next to the "tatami," or mat, where the combats are held. Usually two weight divisions are scheduled for fighting each day of the competition, and the matches of each category are alternated in a two-tatami arena. The finals and semifinals are commonly conducted singly in one of the two combat areas.

PHASE E
Referee Selection

This module was a late addition to the system, and it was conceived, designed, and implemented by Mrs. Evelyn Osako, a programming supervisor at Parke-Davis, the wife of John Osako, Director of Refereeing for the IJF. The program automates the process of referee selection in real time through the application of an algorithm satisfying several key criteria. There is usually a pool of between fifteen to twenty referees, from which teams of three are picked for each match. Usually two matches are fought simultaneously, making the selection process more complex, since each team must conform to three specific rules. The first rule is neutrality. No referee may be from the same country—and, if possible, not even belong to the same

Figure 6
Final Standings

COUNTRY	WON	LOST	AVG.	AVG. TIME PER FIGHT MIN.	TOTAL TIME OF COMBATS	TOTAL MATCHES	WON BY IPPON	WEIGHTED AVG.	GOLD	SILVER	BRONZE	COMPETITORS
ARGENTINA	6	6	.500	3.75	45.02	12	25.00%	.417				6
AUSTRALIA	2	4	.333	5.28	31.67	6	0.00%	.250				4
AUSTRIA		4	.000	4.18	16.71	4	0.00%	.000				3
BELGIUM	8	9	.471	3.35	56.88	17	23.53%	.397				7
BRAZIL	3	10	.231	3.04	39.58	13	7.69%	.192				8
CANADA	7	13	.350	2.59	51.87	20	30.00%	.325				12
CHILE	1	9	.100	2.07	20.71	10	10.00%	.100				9
CHINA	6	6	.500	3.38	40.56	12	25.00%	.438				4
COSTA RICA		4	.000	1.38	5.50	4	0.00%	.000				3
DOMINICAN REP.		3	.000	.57	1.72	3	0.00%	.000				2
FRANCE	20	14	.588	4.01	136.17	34	38.24%	.537				10
GERMAN DEM REP.	4	5	.444	4.89	44.04	9	22.22%	.389				4
GERMAN FED REP.	21	16	.568	4.66	172.36	37	27.03%	.493		2		9
GREAT BRITAIN	3	6	.333	3.51	31.63	9	22.22%	.306				5
ITALY	2	4	.333	4.89	29.32	6	16.67%	.292				3
IVORY COAST		4	.000	1.55	6.18	4	0.00%	.000				4
JAPAN	64	9	.877	4.09	298.81	73	58.90%	.798	6	3	3	12
KOREA	17	14	.548	4.04	125.11	31	35.48%	.500			3	9
LUXEMBOURG		1	.000	.53	.53	1	0.00%	.000				1
MEXICO	7	12	.368	4.05	76.97	19	21.05%	.329				12
MOROCCO	1	1	.500	6.00	12.00	2	0.00%	.375				1
NETHERLANDS	21	17	.553	3.44	130.81	38	26.32%	.454		1	2	10
NEW ZEALAND		2	.000	.32	.64	2	0.00%	.000				2
NICARAGUA		1	.000	2.08	2.08	1	0.00%	.000				1
PANAMA		3	.000	1.60	4.80	3	0.00%	.000				3
PHILIPPINES		2	.000	2.70	5.40	2	0.00%	.000				2
POLAND	3	3	.500	4.46	26.73	6	33.33%	.458				3
PUERTO RICO		8	.000	2.82	22.53	8	0.00%	.000				8
SWITZERLAND		2	.000	3.13	6.25	2	0.00%	.000				2
TUNISIA		2	.000	2.29	4.58	2	0.00%	.000				2
UNITED STATES	9	15	.375	3.83	91.83	24	29.17%	.344				12
U. S. S. R.	19	12	.613	3.89	120.61	31	41.94%	.565			4	8
YUGOSLAVIA	7	10	.412	3.53	60.00	17	17.65%	.338				7
VENEZUELA	1	1	.500	5.11	10.22	2	50.00%	.500				1

JUDOKAS	COUNTRIES	MATCHES	WON BY IPPON	AVG. MATCH	TOTAL TIME
189	34	464	140 = 30.17%	3.73 Min.	1729.82 Min.

continental union—as either of the two combatants. The second rule concerns communications compatibility. That is, the members of each team must be able to talk among themselves through some common language. And the third rule, there must be a minimum level of combined experience in a team—this requirement goes up in importance for semifinal and final matches. In addition to these rules, the program must also take into account that it is highly desirable that all referees officiate about the same number of times in any one tournament.

PHASE F
Log of Combats Generation and Checking

Upon completion of the last match of each weight category, the stage is set for producing all reports and analyses pertinent to the activities within that weight category. Before proceeding to execute these jobs, however, there must be a thorough edit to insure correctness in the final results. The hectic atmosphere inside the contest area, with the wild cheering of the fans, the euphoria of the winners, and the disappointment of the losers, makes it quite possible for the coder to err, either when inputting directly to a terminal or when transcribing to the specific form. In any case, the cross-checking procedure, utilizing basic controls such as the elimination chart, hash totals, and a log of combats, serves as a good safeguard. The log of combats is a report produced from the records of all fights coded and gives a total description of the outcome of each match. (A facsimile from a log-of-combats report is shown in figure 2.)

PHASE G
Generation of Final Reports

In addition to the log of combats, which remains as a basic document accounting for all encounters in an almost descriptive format, two more reports are produced which yield a considerable amount of interest in the "shiai." These are the final weight-division breakdown by competitors and the tabulation of usage frequency for the different judo techniques. Both of these reports are produced at the end of each category's combats and at the completion of the checking process. The first of the reports mentioned (see figure 3) features a detailed breakdown by judoka of the number of wins or losses, won-lost average, mean and cumulative times of combat, performance points, medal won (if any), and a tally of victories and defeats by decision types. The second report highlights the specific moves classified by "wazas" with which any points or half-points were scored. (Waza can be translated from the Japanese as technique, and all judo movements fall into some category of waza. For instance, osoto-gari, big outer reap, is a leg technique, or "ashi waza"; harai-goshi, sweeping hip, is a hip technique, or "koshi waza." For a more complete description of the topic see *Illustrated Kodokan Judo,* Kodansha, Tokyo, 1955.) Both these documents are final for the weight division concerned.

At the conclusion of the championship, all combat records are collected and sorted into various classifications for the production of varied reports. The same holds true of the judoka files and those with country information and totals. Among all the documents produced, five deserve to be mentioned here as being of primary importance. These are: (1) final standing by country; (2) won-lost table for each country; (3) breakdown of all combats by judoka within each country; (4) techniques scored with and scored against by country; and (5) total tally of frequency of utilization of techniques by wazas. (Reports 4 and 5 are shown in figures 4 and 5, respectively, and report 1 in figure 6.)

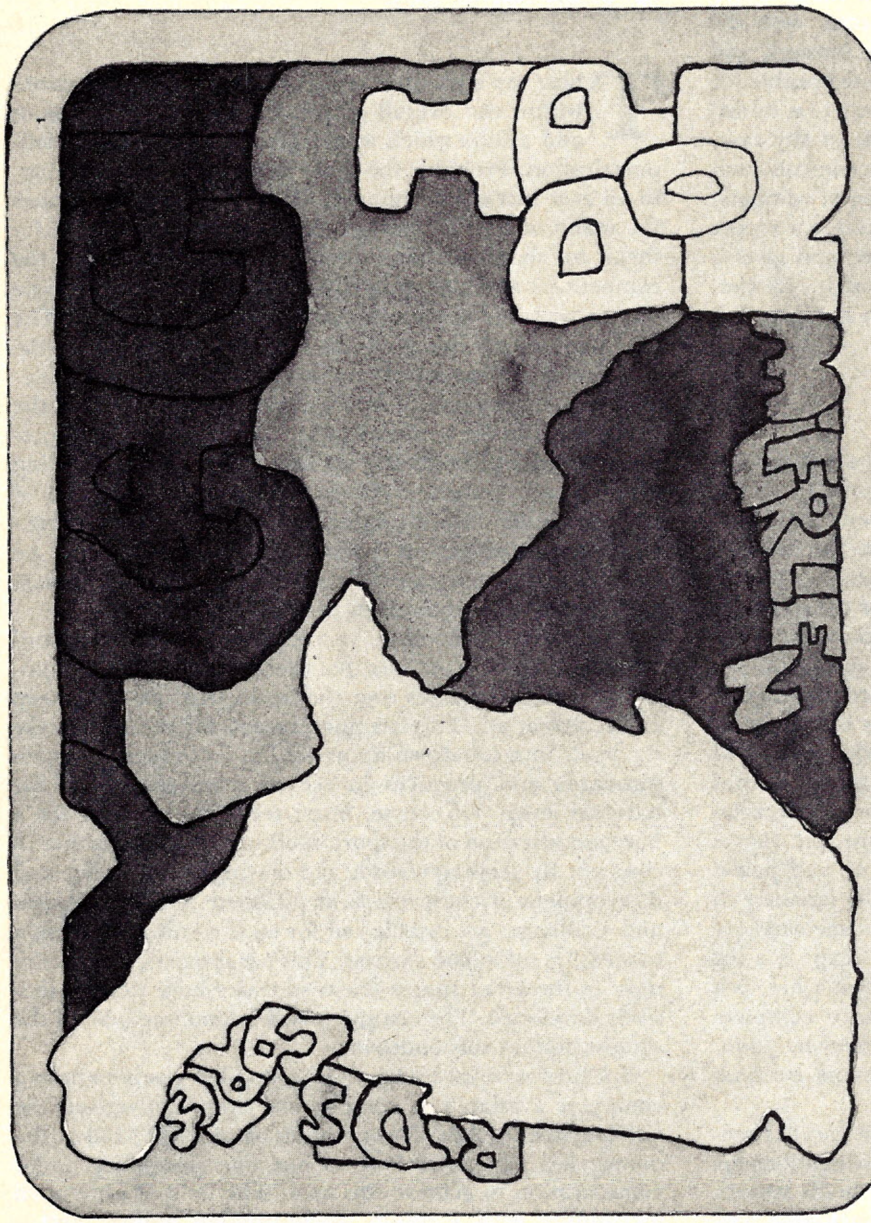
In addition, indexes of competitors and techniques are generated for future retrieval from the different reports. These have proved quite practical when working with the bulk of documents for study and analysis. Of course, computerized retrieval through an interactive system would be much more efficient, and more worthwhile things could be accomplished. But the costs involved in such a system are presently not within the reach of the International Judo Federation. Though with the development of increasingly sophisticated and inexpensive hardware, this will hopefully not be the case for long.

There are a number of quite interesting applications within the overall structure of judo competitions and affairs which would greatly benefit from computerization. Probably the most pressing need is to bring, in an ever increasing manner, the thrill of the sport into the realm of the fan and spectator—to make it easier for them to share in the emotional involvement of the competitors and the competing teams in much the same way as with the great spectator sports of our times. The first step toward this goal has already been taken by the creation of a basic set of measures, or statistics, for judo. This was accomplished, in great part, by the entry of the computer into the world of judo. The next step, and possibly the most difficult one because of the cost, is to bring into the shiai computer-driven scoreboards, where all of the interesting data, accumulated as part of the record-keeping procedures of previous tournaments, can be passed on to the spectator in such a comparative context as to generate more interest in the fights taking place.

There are, as of now, very few standards for record keeping in judo. It is a fact that there are a very large number of judo federations over the world that probably keep no records at all. This is a good reason for the IJF to take the lead, in a top-down approach, in introducing certain standards and practices in record maintenance at the national levels. Of course, in no way does this call for a bureaucratization of the sport. Nothing could be farther in concept. By keeping records and tracing performance and development of competition at different levels, incentives and challenges are established for each coming generation to measure up to and surpass. This has proved quite true in most of the other sports where statistics have been extensively developed. The computer will undoubtedly be of the utmost help in this endeavor.

But first we must make it easier and less costly to have a computer available at competition sites. Converting the JUDO system to APL for operation on an IBM 5100 or the newer, less expensive 5110 is just one possibility under consideration to achieve this goal. The field is wide open for anyone with a micro, programming ability, and a sports-oriented mind—so why not take a chop at it. ▼

Piecing Together



How can you write programs to help with judo statistics if there are no judokas around? Or worse, if you don't know the first thing about judo? You see, I know a bit about programming, and I have a 5100 in my living room. In my efforts to demonstrate that the JUDO system could indeed be implemented in APL on an IBM 5100, I discovered that there is a great deal to programming certain applications that most programmers don't know. Like what's a *waze?* Or what's an *ippon?* Or any one of a myriad of other funny-sounding concepts that seem to be impossible to translate directly into APL.

In fact, that seems to be a major problem with programming. Not only do you have to know how to program, but you have to have a really thorough knowledge of the application—be it JUDO, accounting, or artificial intelligence. Maybe it seems obvious to you that in order to write a program to balance your checkbook, you have to know basically how to do it yourself, even if you're not handy with debits and credits. The same is true for pretty much every application I've seen. But it goes deeper than that. In order to program with any ease, you have to know the specifics of what you're programming. In other words, in order to write a program to balance your checkbook, you have to know what a check is, how much it might be worth, how long most checks remain uncashed, and so on. Knowing the specifics really is important. That's why there are so many computer jocks who go into systems programming. If you know only computers, the best you can do is program a computer to run another program.

a Judo Program

by Eben Ostby

As I learned, the same holds true for the JUDO system. My demonstration not only proved that JUDO can be run on a 5100, but also that a little fudging can make up for a lot of ignorance. Still, it does show basically what's involved in writing a JUDO package.

I chose to implement just a small fraction of the whole system. The programs I wrote handle the basics of record keeping: programs to set up some of the tables with data for each country, each competitor, and each combat; programs for adding new data to these tables; and some (token) report-generating programs.

The setup programs are extremely simple. Since in my version all the data are held in the APL "active workspace," the programs merely create "empty" matrices. An empty matrix is just one with no data in it. In APL, this is done by creating the matrix with no rows. Every time a new item is to be added to the existing data, another row is added onto the matrix. This arrangement also saves space in the computer's memory, since no space is reserved for future data until the time it is actually input.

There are three setup programs: SETCOUNTRY, SETCOMP, and SETCOMBAT. These programs create the empty matrices for data that applies to each country, each competitor, and each combat, respectively. If you can read APL, you'll see that they create the matrices, and then they call a program called ADDCOUNTRY, ADDCOMP, or ADDCOMBAT. That is to get you started on entering the data for each category. For the most part, ADDCOUNTRY and ADDCOMP are pretty similar. First, they ask for the name of the country or the competitor; then they ask for a series of data items. If the user enters nothing in response to any of these items, then the program asks for the *previous* item again. That way, most errors can be avoided or fixed. If the user enters nothing in response to the prompt for the name, then the program terminates. The user can ADD more later, if he wants to.

ADDCOMBAT is similar in idea, although the details are a bit different. Since my knowledge of judo really fell flat when it came to things like techniques and decisions and such, I left the program fairly simple. All responses for these items are in the form of numbers. (When I learn more about what moves and decision types are used, I could then substitute a small APL function that will accept as input the name of a technique and return a number, but for now I've left it this way.) If the user enters NAME for the number of the winner or the loser, however, a little program called NAME will ask for the name of the combatant to be entered; then it will substitute the appropriate number. In ADDCOMBAT, you can also enter STOP or just hit the carriage return; it will start over with the first question again, or, if you are in the process of answering the first question, the program will terminate.

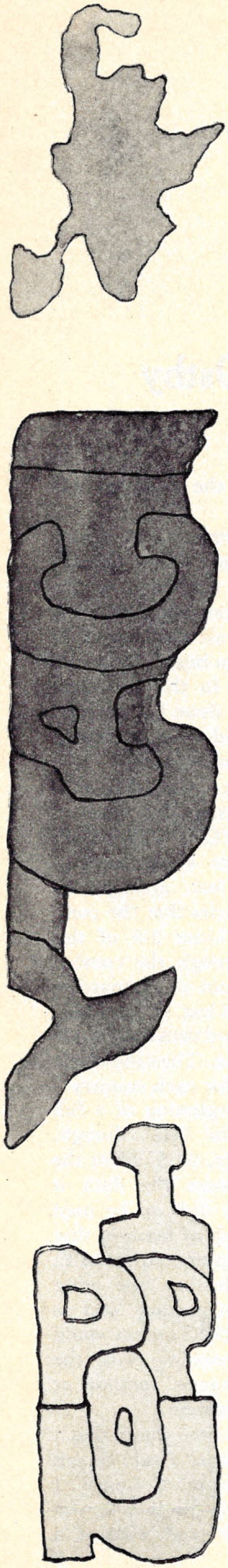
There are loads of features—referee selection, random pairing—that are easily programmed in APL.

There are two very simple report-generating programs. One, COMPSUM, produces a summary of the competitors. In a neat tabular format, this program lists all the names of competitors that you input along with the vital statistics. In this program, most of the work is done by the APL format function, and I just catenated the different columns together.

The other report-generating program gathers statistics by country, hence its name, BYCOUNTRY. BYCOUNTRY provides a summary of matches won and lost, arranged in groups by the countries that the combatants represent. Lines 3-9 of this program search through the combat data matrix to find out which matches each person won and lost. A few statistics are computed, and then on lines 10-16 the data are output. Then the whole process is repeated for each country.

This is just the beginning of a full JUDO system for the 5100, though. For long-term storage, the 5100, or the new 5110, could store the data it gathered on tape or disk, to be used later. There are loads of features that could be easily programmed in APL, too—the referee selection, the other reports, the random pairing, and so on. Further, a real JUDO system would have facilities for working with the rules of the "repechagge" method or the "five bad point system."

Perhaps when the time comes that I know more about judo, I can extend some of these programs. Certainly, I have found that it is possible to run JUDO with a 5100—and without a judoka. ▼



```
      ∇ SETCOUNTRY
[1]    CNTRYΔN← 0 8 ρ0
[2]    CNTRYΔA← 0 2 30 ρ' '
[3]    ADDCOUNTRY
      ∇

      ∇ SETCOMP
[1]    CMPΔN← 0 8 ρ0
[2]    CMPΔA← 0 30ρ' '
[3]    ADDCOMP
      ∇

      ∇ SETCOMBAT
[1]    COMB← 0 4 7 ρ0
[2]    ADDCOMBAT
      ∇

      ∇ ADDCOUNTRY;⎕IO;I;J;K
[1]    ⎕IO←0
[2]   MLOOP:→(0=ρI←ASKA 'ENTER COUNTRY NAME' )/0
[3]    CNTRYΔA←CNTRYΔA,[0](30↑I),[¯0.1]30↑ASKA'NAME OF COACH'
[4]    CNTRYΔN←CNTRYΔN,[0]I←''ρρCNTRYΔA
[5]    J←1
[6]   ILP:→(0=ρK←ASKN TAGY[J-1;])/BK
[7]    CNTRYΔN[I-1;J]←1↑K
[8]    →((8≠J←J+1)/ILP),MLOOP
[9]   BK:→(0≠J←J-1)/ILP
[10]   CNTRYΔA←¯1 0↓CNTRYΔA
[11]   CNTRYΔN←¯1 0↓CNTRYΔN
[12]   →MLOOP
      ∇

      ∇ ADDCOMP;⎕IO;I;J;K
[1]    ⎕IO←0
[2]   MNLP:→(0=ρI←ASKA 'ENTER COMPETITOR''S NAME')/0
[3]    CMPΔA←CMPΔA,[0]30↑I
[4]    CMPΔN←CMPΔN,[0]I←''ρρCMPΔA
[5]    J←1
[6]   ILP:→(0=ρK←ASKN TAGP[J-1;])/BK
[7]    CMPΔN[I-1;J]←1↑K
[8]    →(7≠J←J+1)/ILP
[9]   RECN:K←10↑ASKA 'NAME OF COUNTRY'
[10]   →((1↑ρCNTRYΔA)=K←((CNTRYΔA[;0;⍳10])∧.=K)⍳1)/BK
[11]   CMPΔN[I-1;7]←K
[12]   →MNLP
[13]  BK:→(0≠J←J-1)/ILP
[14]   CMPΔN←¯1 0 ↓CMPΔN
[15]   CMPΔA←¯1 0 ↓CMPΔA
[16]   →MNLP
      ∇

      ∇ ADDCOMBAT;⎕IO;I;J;K
[1]    ⎕IO←0
[2]    COMB←COMB,[0]0
[3]   STAR:→(0=ρJ←⎕,0ρ⎕←'ENTER WINNER ID')/DRP
[4]    COMB[I;0;⍳2]←(1+I←¯1+''ρρCOMB),J
[5]    →(0=ρJ←⎕,0ρ⎕← 'ENTER LOSER ID' )/STAR
```

```
[6]    COMB[I;0;2]←J
[7]    →(∨/J←~COMB[I;0;1↓ι3]∊1+ι1↑ρCMPΔA)/TIM
[8]    (,⊖J⌿ 2 6 ρ 'WINNERLOSER ' ), ' ID IN ERROR.'
[9]    →STAR
[10] TIM:→(0=ρJ←ASKN 'TIME (MINUTES SECONDS)' )/STAR
[11]   COMB[I;0;3]← 60 60 ⊥2↑J
[12]   →(0=ρJ←ASKN 'DECISION TYPE' )/STAR
[13]   COMB[I;0;4]←1↑J
[14]   →(0=ρJ←ASKN 'NO. OF YUJOS KOKAS' )/STAR
[15]   COMB[I;0;5 6]←2↑J
[16]   COMB[I;1;]←7↑ASKN 'WINNER''S TECHNIQUES'
[17]   COMB[I;2;]←7↑ASKN 'LOSER''S TECHNIQUES'
[18]   COMB[I;3;]←7↑ASKN 'PENALTIES'
[19]   →STAR-1
[20] DRP:COMP← ¯1 0 0↓COMB
       ∇

       ∇ R←ASKA T
[1]    R←(ρT)↓⍞,0ρ⍞←T←T, ': '
       ∇

       ∇ R←ASKN T
[1]    R←¯1↓⍎((ρT)↓⍞,0ρ⍞←T←T, ': ' ), ',1'
       ∇

       ∇ COMPSUM
[1]    (30↑ 'NAME' ), '        ID  HEIGHT  WEIGHT     AGE    RANK   EXP
WT.DIV.   COUNTRY'
[2]    100ρ'='
[3]    CMPΔA,( 8 0⍕ 0 ¯1 ↓CMPΔN),' ',CNTRYΔA[CMPΔN[;7];0;]
       ∇

       ∇ BYCOUNTRY
[1]    A←⎕IO←0
[2]    M←(CMPΔN[;7]=A)/CMPΔN[;0]
[3]    M←M,(+/M∘.=COMB[;0;1]),[0.1]+/M∘.=COMB[;0;2]
[4]    M←M,M[;1]÷+/M[;1↓ι3]
[5]    M←M,(+/(J←M[;0]∘.=⍉COMB[;0; 1 2 ])×((ρM)[0],2×(ρCOMB[0])ρCOMB[;0;3])÷60
[6]    M←M,M[;4]÷+/M[; 1 2 ]
[7]    M←M,(¯1 TYPS 1),[0.1]¯1 TYPS 2
[8]    M←M,(M[;1]-+/M[;6 7 ]),(1 TYPS 1),[0.1]1 TYPS 2
[9]    M←M,M[;2]-+/M[; 9 10 ]
[10]   CNTRYΔA[A;0;]
[11]   ''
[12]   (30↑'JUDOKA' ),'WON LOST AVG. TIME AV.TIME IPPON Y-GACHI OTHER
IPPON Y-GACHI OTHER'
[13]   CMPΔA[M[;0]-1;], 3 0 5 0 5 2 6 2 7 2 6 0 8 0 6 0 6 0 8 0 6 0⍕0 1↓M
[14]   2 1ρ' '
[15]   'TOTAL MATCHES ',(⍕+/,M[; 1 2]),' WON BY IPPON ',(7 2⍕100×(+⌿M[;6]
)÷+/,M[; 1 2 ]), ' ∘/∘'
[16]   ''
[17]   →((1↑ρCNTRYΔA)≠A←A+1)/2
       ∇

       ∇ R←DV TYPS T
[1]    R←+/T=(((0,DV)×(ρCOMB)[0])↓J)×((ρM)[0],(ρCOMB)[0])ρCOMB[;0;4]
       ∇
```

EXAMPLE OF RUNNING SETCOUNTRY

```
      SETCOUNTRY
ENTER COUNTRY NAME: USA
NAME OF COACH: THURBER
  SIZE OF DELEG: 20
   NO. OF DOJOS: 401
    BLACK BELTS: 100
  YEARS OF JUDO: 6
     POPULATION: 300000000
     COACH RANK: 4
   COACH EXPER.: 7
ENTER COUNTRY NAME: GRAND FENWICK
NAME OF COACH: HORACE
  SIZE OF DELEG: 3
   NO. OF DOJOS: 2
    BLACK BELTS: 6
  YEARS OF JUDO: 4
     POPULATION: 5002
     COACH RANK: 90
   COACH EXPER.: 4
ENTER COUNTRY NAME: UPPER VOLTAGE
NAME OF COACH: TOM
  SIZE OF DELEG: 55
   NO. OF DOJOS: 13
    BLACK BELTS: 43
  YEARS OF JUDO: 4
     POPULATION: 10
     COACH RANK: 54
   COACH EXPER.: 3
ENTER COUNTRY NAME: ....
```

EXAMPLE OF SETCOMP

```
      SETCOMP
ENTER COMPETITOR'S NAME: B. ARNOLD
         HEIGHT: 72
         WEIGHT: 103
            AGE: 24
           RANK: 15
     EXPERIENCE: 2
WEIGHT DIVISION: 9
NAME OF COUNTRY: UPPER VOLTAGE
ENTER COMPETITOR'S NAME: M. FURLIEN
         HEIGHT: 68
         WEIGHT: 155
            AGE: 31
           RANK: 15
     EXPERIENCE: 3
WEIGHT DIVISION: 9
NAME OF COUNTRY: USA
ENTER COMPETITOR'S NAME: E. MCDONALD
         HEIGHT: 210
         WEIGHT: 97
            AGE: 11
           RANK: 31
     EXPERIENCE: 3
WEIGHT DIVISION: 2
NAME OF COUNTRY: GRAND FENWICK
ENTER COMPETITOR'S NAME: R. DECARTES
         HEIGHT: 402
         WEIGHT: 441
            AGE: 53
           RANK: 90
     EXPERIENCE: 2
WEIGHT DIVISION: 2
NAME OF COUNTRY: GRAND FENWICK
ENTER COMPETITOR'S NAME: ....
```

EXAMPLE OF SETCOMBAT

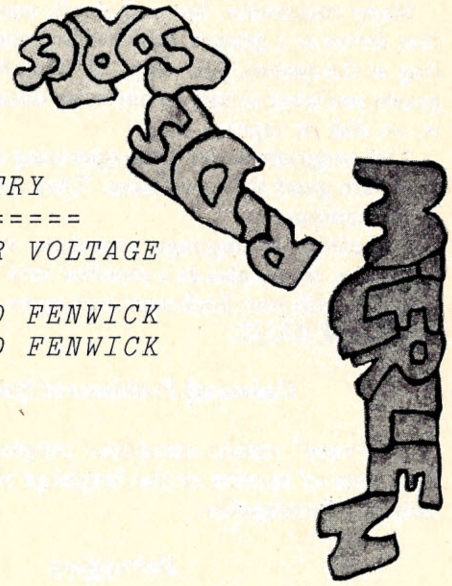
```
      SETCOMBAT
ENTER WINNER ID
⎕:
```

```
                 COMPSUM
NAME
===============
M. FURLIEN
E. MCDONALD
H. TRUMAN
R. DESCARTES
```

```
      BYCOUNTRY
USA
JUDOKA
E. MCDONALD

TOTAL MATCHES 2 WON BY IPPON

GRAND FENWICK
JUDOKA
H. TRUMAN
R. DESCARTES

TOTAL MATCHES 3 WON BY IPPON

UPPER VOLTAGE
JUDOKA
M. FURLIEN

TOTAL MATCHES 3 WON BY IPPON
```

```
         4
ENTER LOSER ID
⎕:
         1
TIME (MINUTES SECONDS): 2 51
DECISION TYPE: 1
NO. OF YUJOS KOKAS: 1 1
WINNER'S TECHNIQUES:
LOSER'S TECHNIQUES: 2
PENALTIES:
ENTER WINNER ID
⎕:
         1
ENTER LOSER ID
⎕:
         2
TIME (MINUTES SECONDS): 4 31
DECISION TYPE: 2
NO. OF YUJOS KOKAS: 1 0
WINNER'S TECHNIQUES:
LOSER'S TECHNIQUES: 5
PENALTIES:
ENTER WINNER ID
⎕:
         NAME
ENTER NAME: R. DESCARTES
ENTER LOSER ID
⎕:
         2
TIME (MINUTES SECONDS): 2 13
DECISION TYPE: 20
NO. OF YUJOS KOKAS: 0 0
WINNER'S TECHNIQUES:
LOSER'S TECHNIQUES: 1.
PENALTIES: 3 5 1
ENTER WINNER ID
⎕:
         STOP
```

COMPSUM Printout



ID	HEIGHT	WEIGHT	AGE	RANK	EXP	WT.DIV.	COUNTRY
1	68	155	31	15	3	9	UPPER VOLTAGE
2	210	97	11	31	3	2	USA
3	4	4	4	4	4	4	GRAND FENWICK
4	402	441	53	90	2	2	GRAND FENWICK

BYCOUNTRY Printout

WON	LOST	AVG.	TIME	AV.TIME	IPPON	Y-GACHI	OTHER	IPPON	Y-GACHI	OTHER
0	2	.00	6.73	3.37	0	0	0	0	1	1

.00∘/∘

WON	LOST	AVG.	TIME	AV.TIME	IPPON	Y-GACHI	OTHER	IPPON	Y-GACHI	OTHER
1	0	1.00	10.92	10.92	0	1	0	0	0	0
2	0	1.00	5.07	2.53	1	0	1	0	0	0

33.33 ∘/∘

WON	LOST	AVG.	TIME	AV.TIME	IPPON	Y-GACHI	OTHER	IPPON	Y-GACHI	OTHER
1	2	.33	18.28	6.09	0	1	0	1	1	0

.00 ∘/∘

Missionary Position

PROGRAMMING LIKE A NATURAL

by Theodor Nelson

The Program You See Most Of

Many computers, big and small, come with a program that serves as a general butler of the computer system. Sitting at the system, you ask it to bring forth whatever programs you want to use, or put away data in the closet (that is, on disk or tape).

This program butler is the operating system, or monitor. They are good things to have. They are offered for many dinky computers.

Sometimes a language processor, such as the BASIC processor, serves also as a monitor and will store data and edit files with you. Such monitors come with most amateur versions of BASIC.

Universal Problems of Software

"Software" means computer programs. Regardless of your area of interest or the language you use, some questions are inescapable.

Debugging

It is natural to make mistakes while you are programming. Some people get better and better at programming, and make fewer and fewer mistakes. However, the mistakes anybody makes can be awfully big ones.

Mistakes in programming, also called bugs, are not easy to find. Surprisingly, it is impossible to tell by looking at a computer program whether it will work or not. The only way to test a program, except in a small number of mathematical cases, is to try the program and see if it works. Indeed, a program may work correctly at one time and yet have hidden bugs that may make it fail later on.

As the amount of known territory increases, the amount of unkown territory increases even faster.

The problem gets worse as programs get bigger. Ordinarily a medium-sized program does not work the first time. Or the second. Or the tenth. But the human creating this program, struggling to find his omissions and mistakes, perfects small pieces of it at a time. And with the perfection of each piece, gets a sense of drawing closer to the overall goal.

The complications of computer programming were not obvious at the start. Henry Tropp, who has done a research project on the history of computing, interviewed the man who discovered debugging, an English scientist. He wrote a program for a computer of the 1950s and discovered that the program did not run correctly. He found one of the errors, changed it, and discovered that the program still did not run correctly. With sinking heart it occurred to him that he would spend the rest of his working life "attempting to correct my own mistakes."

The programmer subsists on piecemeal reward, sometimes a little reward for a lot of effort, sometimes a great reward for a little effort, sometimes seemingly no reward at all. Yet this intermittency of reward, and the rare grand feeling you get when it works, seem to be enough to keep great numbers of people hard at work in programming activity. (Behavioral psychologists are quick to tell us that intermittent reward is the kind that promotes learning most effectively. But what may be more important is the good feeling when the program works.)

Can a Thing Be Done?

We get lots of ideas for things to do with computers; but not every idea is doable.

A very serious problem for the beginner is not knowing what constitutes an undoable problem, or one which is just too big. The beginner, successful with a small project, rushes right on to attempt the impossible, rushing in where experts fear to tread. (But it is just through the fearlessness of the newcomer—the kids who know no fear or modesty—that many important innovations occur.)

Structured Programming

A new set of rules is having great impact. "Structured programming" is a set of rules for writing programs that are easier to debug, cheaper to produce, easier to improve or fix up. Basically, structured programming means dividing the programs into blocks of certain kinds, which behave and interrelate in certain ways. The rules are just a hair too complex for this article.

Structured programming has become a sort of religion in recent years, spread by its founder, Edsger Dijkstra of the Netherlands; by Harlan Mills within IBM; and many others, notably Henry Ledgard, author of *Programming Proverbs,* and Brian Kernighan, author of *The Elements of Programming Style.*

BASIC, and some of the other traditional computer languages, are difficult to use according to the rules of structured programming. This is beginning to look like a strong argument for the Lambda languages and certain others; they make it possible to get your programs running faster and change them more readily.

The Complications Go On and On

Mankind is just learning what the consequences and complications of such plans of operation—detailed computer programs—really are. In the twenty years since programming began, it has been studied extensively. A great deal has been learned within the field about how programs work, and even more has been discovered that is confusing and unknowable. As the amount of known territory has increased, the amount of unkown territory has increased even faster. Programming is still an art, not a science.

Each small step forward has revealed the immensity of the unknown void beyond, just as astronomy in the twentieth century has shrunk mankind faster and faster in an unthinkably large universe.

Software Quagmires

It is all too easy to keep trying to fix programs that were really very bad in the first place and throw good money and effort after bad. What is worse, other people have to use the quagmire that is thus created. (IBM, indeed, is notorious for their cumbersome and sprawling software—but if the customer is locked in, IBM profits from the inefficiency of that software.)

Yet, like the Vietnam War, software can become justified simply on the grounds that too much has been invested in it already.

It is best to take the advice offered in *Programming Proverbs* by Henry Ledgard. "Don't be afraid to start over."

Programs and Program Packages

Since the 1950s, computer programs have been valuable objects of sale. Programs have been sold for wide varieties of purposes—usually for business, but also for science and government.

The price of individual program packages has always been, of course, what the market would bear. It is not uncommon for a language processor to cost tens of thousands of dollars to a user organization. Application programs—for specific business uses on large computers—can also cost tens of thousands of dollars. Programs may be rented instead, in which case the monthly payments can be very, very high.

This has been the world of software. The little computers, though, should have a drastic effect on the price and style of software. Right now nobody quite knows *what* effect. What is going to happen with software in the amateur market is a mystery, but we can expect the price to go down for businesses. The price of good programs for personal users may go up into the hundreds. (Thousands???)

Depending on what hardware becomes popular, programs may be sold in little wafers, or sticks like chewing gum, or cubes, all plugging into the computer somehow.

And some will be sold as they already are—on cassettes and paper tape and disks. All these are merely forms of storage for the programs: the series of commands that run the computer. But because programming is hard work, the programs may be sold as objects of value.

The principal software for the personal market will consist of canned interactive systems for an ever-widening

spectrum of purposes, and in a growing range of styles. The programs for your home computer will not merely be sold singly. They will also come in suites, that is, integrated collections of programs that fit together. (We may even look forward to panoramic software—linked programs for a broad spectrum of personal uses.)

We will discuss the legal matter of software protection later. But whatever the outcome there is to this legal issue, there will surely evolve a stable fashion by which developers of good programs can receive financial reward for them.

Can You Get the Program—and Is It in Your Style?

You can do anything with your computer that you have a program for. If you buy a canned or prepared computer system for some purpose, you do not have to learn to program. .... Most personal computer applications are going to use software somebody else has developed.

It is easy to make a perfect copy of much of the software for little computers.

Now, either the program exists, or it doesn't. But just because programs for a given purpose exist does not mean they are any good.

There is usually considerable leeway in how a program can be designed. Programs that supposedly do the same thing can be as different as hats, or dogs. Many write-ups on home computing in the popular presses might give the impression that the computer will do whatever you want, in the style you expect, with someone else's program. This is almost never true. You will have to adapt to another's idea of what aspects are important, and how they are best explicated in the program. Even if a program like the one you wanted already exists, it probably is not in the style you would like. And if it does not exist, you are going to have to create it. One's personal fantasies, often so clear, tend not to be what the other guy programmed. (Great disappointments occur.) Each person's preferred style of use may be different from another's.

Unless you are the one who programs it, it will not be focused as you would have it, nor as flexible in the ways you might want.

If you are going to use a pre-existing program, you have to adapt to *it*. Otherwise you must program it yourself or adapt the pre-existing program. Thus you must learn how to program. The same goes for many of these new applications, you are either going to have to program yourself, or have somebody else do it for you. "If you want a thing done right, do it yourself." The way you design it is crucial. Can it be made easy to use? Making things easy for people is hard. But it can be done. You have to try hard enough and be able to visualize. (See "More Art of the Computer Screen," by Theodor Nelson in the February 1978 *ROM*.)

Selling Software as Hardware

Programs are being sold on paper tape and cassette. When loaded, their contents slide into the otherwise empty spaces of the machine. When you're done, you obliterate the old program and use the computer's memory for something else.

However, programs will also be sold by some manufacturers as little plug-in thingies. "Thingies" is a vague term, but these plug-in programs can come in any size and shape. Some are now sold, not for computers but for calculators, in little wafers the size of sugar cubes.

These are ROMs—Read-Only Memories. These little memories, filled with their programs, behave just like the regular changeable memories of the computer when they are temporarily loaded with a program. But the ROMs are permanent.

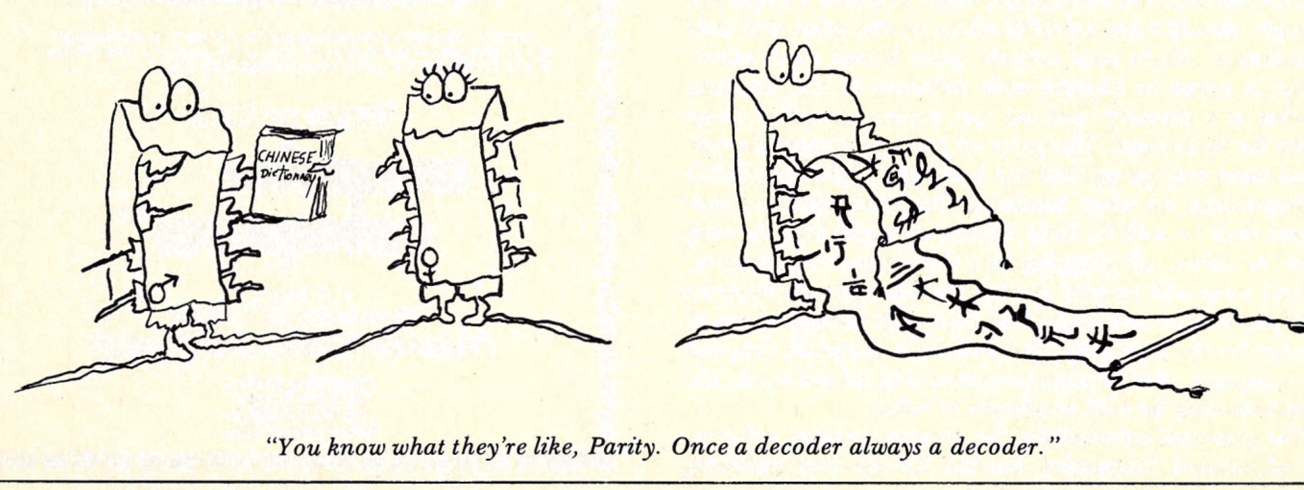
There is no real logic distinction between one type of program and another. But the ROMs are more convenient—and people are perhaps less likely to copy the programs that are on them.

But this is not clear. Let us consider, at this point, steps that can be taken to enforce the ownership and salability of programs.

Program Protection

Most amateur computers can presently use each other's programs. By law, the owner may charge anyone who

EVE'N'PARITY



"You know what they're like, Parity. Once a decoder always a decoder."

wants a copy of a program he has developed—but in fact, one hobbyist may easily give a copy to another on the sly. This is the copyright problem. There has been a great deal of program copying by hobbyists in the last couple of years. Nobody knows how much, and of course nobody—except a few troublemakers—is going around bragging that he has done this.

It is easy to make perfect copy of much of the software for little computers.

Herein lies the temptation.

Business users pay readily for software, since it is an obvious business expense.

However, amateurs who only "want to have fun" are hard to persuade that making a copy is as dishonest as counterfeiting a dollar bill. In the next few years, however, it will become clear how much most people will depend on programs that are developed by others, and how very much better some of them are than others. This will affect people's thinking on the issue.

Intellectual Property

Just as background, let us review the main ways that United States law allows you to own something you come up with in your mind. A lot of people seem to think you can patent or copyright anything. This is far from the truth.

(Note that these laymen's descriptions should not be taken as a legal guide. Consult a lawyer for the exact information—and the latest. Things are changing fast.)

The law provides several methods by which people are granted certain rights to things they make up.

Patents

The most famous of these is the patent. The patent is expensive to get, may not be binding, and lasts for only seventeen years. It protects your invention only in the narrowest sense: with reference to certain specific features which nobody can copy without your permission.

The patent was established by Congress with the stated intent of encouraging the communication of technical knowledge. For this reason, it must describe fully what is being covered. In return for this description, the government gives the inventor exclusive rights to the invention—in the narrow sense covered by the wording on the actual patent—for the seventeen years.

To patent something, you must search to see what is already patented, or known, that is like it. If you think yours is original, you must submit a patent application. Your attorney argues with patent examiners for months or years, then maybe you get it and maybe you don't.

The expense of getting a patent is generally several thousand dollars—in part based upon the attorney's judgment of your willingness to pay, in part on the complexity of your patent application. But such sums are usually out of reach for people trying to start a business on a shoestring, as most of the people interested in this matter are.

Furthermore, there is some considerable doubt as to whether patents can be obtained for computer programs. The Supreme Court has ruled lately that programs by themselves are not patentable, but that clears up less than some people think.

Cryptic Computer

CODING COMPUTER DEVELOPMENT

by Frederick W. Chesson

In last month's column, the contributions of World War II technology to the foundations and growth of digital computers were discussed. Now, a publication released by the National Security Agency, the nation's top-secret cryptographic establishment, reveals how the infant computer industry was nurtured by the requirements for national security.

"Influence of U. S. Cryptographic Organizations on the Digital Computer Industry" is a modest thirty-eight page booklet, dated May 1977, written by Samuel S. Snyder. To put this booklet together, the author was faced with the task of examining and digesting a host of once highly classified documents, dating back a good thirty-five years.

Before the National Security Agency's establishment by President Truman's Executive Order of 4 November 1952, the Armed Forces Security Agency, formed in 1949, had tried to unify the competitve Army, Navy, Air Force, and State Department cryptographic organizations. In the pre-computer era, circa 1935, The Army Security Agency and the Navy's Communications Supplementary Activities, Washington (CSAW) had begun to expand upon the basic IBM punch-card tabulating machines. CSAW was especially active in having commercial firms design and build increasingly specialized apparatus, employing high-speed (for relays) logic circuits. Besides IBM, such well-known companies as Bell Laboratories, Eastman Kodak, and National Cash Register were involved when World War II burst upon the scene.

After the war, a group of Naval officers, familiar with the problems of secrecy in research and manufacturing, set up their own organization, Engineering Research Associates (ERA), to develop and build new equipment for CSAW. In 1947 they pioneered the magnetic storage drum, once the mainstay of computer memory. One early model rotated at 225 rpm, with signals read, erased, and rewritten on the same track at the then unheard-of rate of 20,000 pulses-per-second. By 1948, drums thirty-four inches in diameter were rotating at 3,500 rpm for a computer system aptly named DEMON.

In July and August of 1946, a series of lectures was held at the University of Pennsylvania's Moore School of Engineering, at Philadelphia. The infant computer giants, EDVAC and ENIAC, were the subject of the symposium, which attracted both government and private representatives. Among the speakers were such illuminaries in the galaxy of the new art as John von Neumann, J. P. Eckert, J. W. Mauchly, Herman Goldstine, and Howard H. Aiken. In this stimulating atmosphere, the possibilities of a machine capable of attacking a far wider range of problems than its specialized ancestors were made apparent.

Lt. Commander Pendergrass, CSAW's representative, appreciated these new vistas so much that, within a few months of the symposium, negotiations between the Navy and ERA resulted in a proposal for the development of ATLAS—a computer named for the intellectual giant of small stature in the comic strip "Barnaby." About a year later, a similar report was submitted to the National Bureau of Standards, proposing a similar computer for the Bureau's postwar scientific research programs.

The logic design of ATLAS was influenced by a machine built by the Institute for Advanced Study at Princeton University and by M.I.T.'s own WHIRLWIND. All three of these machines were of the one-address type, although ATLAS employed a different word size. Its memory was an improved drum, having a seventeen-millisecond access time, with a capacity of over 16,000 words of twenty-four bits. A later improvement, called "interlace plugging," reduced access time still further to a minimum of only thirty-two microseconds. ATLAS was delivered in December 1950 and was followed by a second unit in March 1953. They were followed by ATLAS II, which came in two models (one of them had the first core-type memory for a commercial machine). The ATLAS instruction code had forty-one commands, including such instructions as CLEAR ADD, SUBSTITUTE DIGITS, SPLIT CLEAR ADD, and ZERO-CONDITIONAL JUMP.

Cryptogram: ANDES PARTS PHELD BUGRA FEPID SBFER LEAND ETOOL UGRSI ETOEP ENTLA ETOER SWDUD SWDSA EPSWE TOADS EBTCU PARSI EPGCF ESPZF EPSWE TANDG EITZ D GSCDS APHET GIPSR JPART SLEWD PHRSI EMRON EBGFU ATLFL ADCLE.

Hint: one letter used for word space. ***Solution on page 108.***

Not to be outdone by CSAW's quantum jumps, the Army's ASA was also active, considering proposals from Raytheon, Sperry Rand, and ERA as well. With assistance from the NBS, they commenced work on their own computer, employing mercury delay-line memories, under the codename ABNER. By April 1952, ABNER was operational and was followed in June 1955 by a second model, built entirely by the commercial firm, Technitrol. Its instruction code of thirty-one operations included COMPARISON COUNT, EXPAND TRANSFER, RANDOM JUMP, and SWISH. (This latter operation simulated the action of a high-speed comparator, an important function, considering the role data comparison played in the statistical aspects of codebreaking.)

Even uncompleted projects, such as NOMAD, were helpful to the civilian computer industry (for instance, the

entry of Minneapolis-Honeywell into the data field via an arrangement with Raytheon, which had been burned in the NOMAD development).

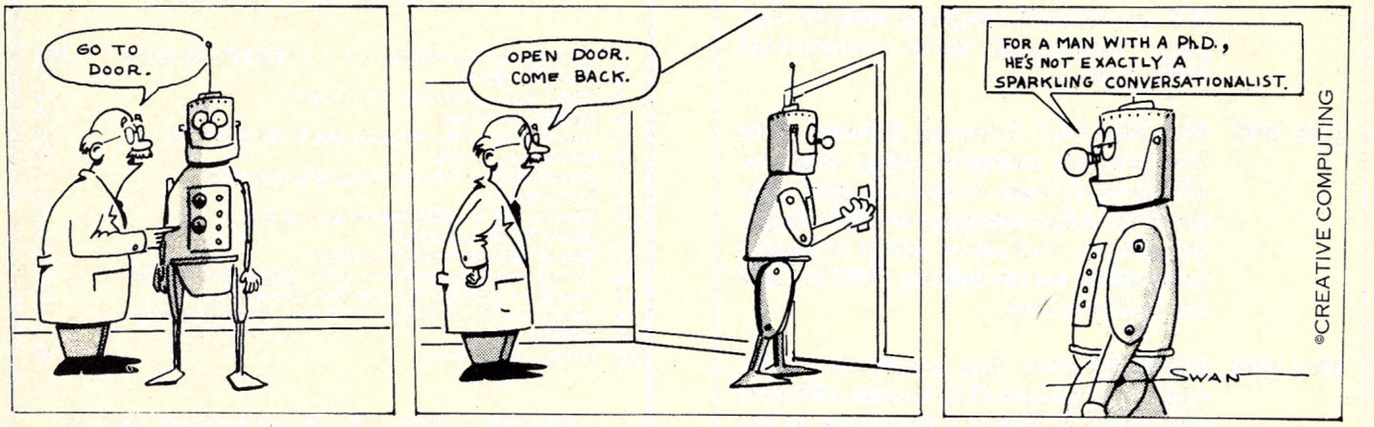
Punch-tape machines developed by IBM during the war led eventually to the Model 702, the first IBM computer designed for information processing. Starting in 1957, NSA established a laboratory for the testing of the many magnetic tapes then on the market, resulting in a draft of industry standards which was circulated in 1966. This action stimulated competition among the manufacturers, reducing the price of a reel from a maximum of $150 to $12 by 1967, with substantial declines thereafter.

The mid-1950s saw an increasing rush of technology, along with a flood of Cold War-generated crypto data, into NSA. It was soon realized that too much time was being spent with data conversion, formatting, and editing information to make it ready for the big machines. So, out of this requirement for speed came BOGART—named, not after actor Humphrey, but in honor of John B. Bogart, famous city editor of the *New York Sun* newspaper. Five machines were ultimately built by ERA. One was modified into ROB ROY, a remote-controlled system with five terminals. Its development was reflected in the UNIVAC 490 system and Control Data's CDC 160 and CDC 1604, while the IBM type-727 tape drives were destined to become industry standards. Drum storage gave way to diode-core memory with a twenty-microsecond cycle time.

The attractiveness of remote terminals became increasingly important, as NSA underwent constant expansion to meet Cold War tensions. Late in 1957, a thrity-five-million-dollar complex was completed at Fort Meade, Maryland, half way between Baltimore and Washington. This centralized many, but hardly all, of the Agency's operations which had been scattered about the capital area in places such as Arlington Hall, a former exclusive school in the Virginia hunt country. Employees of NSA then numbered close to ten thousand. By the mid-1960s, the staff was estimated to exceed fifteen thousand, including those stationed about the world at sensitive listening posts from Tiawan to Turkey. In 1963, an eleven-million-dollar contract was awarded to construct a nine-story "Annex" at Fort Meade.

By the mid-1950s, the transistor was seriously replacing the vacuum tube. In June 1955, the Philco Corporation was awarded a contract for SOLO, an all-transistor, individual desk-sized computer for multiple work areas. Although technical problems in the state-of-the-art development limited the total number of NSA purchases, Philco was sufficiently encouraged to proceed with its own commercial versions, designated TRANSAC S-1000 and PHILCO S-2000.

IBM became increasingly involved with NSA computers in the form of such highly advanced projects as STRETCH, HARVEST, and FARMER. The Atomic Energy Commission, whose requirements were for high-speed multiplication rather than data manipulation, also received a STRETCH model, thus demonstrating the system's versatility and commercial potentialities. HARVEST was enhanced by TRACTOR, a data storage subsystem which could seek out and automatically thread over 150 tape cartridges, storing almost *ninety billion* characters. Such



NSA Computer "Firsts"

Oct 1948	DEMON placed in operation. First practical use of magnetic drum for data storage for analytic operations at electronic speeds.
Dec 1950	ATLAS I delivered; operational in one week. First parallel electronic computer in U.S. with drum memory. Forerunner of commercial E.R.A. 1101.
Apr 1952	ABNER operational; designed and built at NSA. Serial computer similar in logic to SEAC and EDVAC. Most sophisticated computer of its time. First use of computation simultaneous with input-output. Most complete complement of input-output capabilities (punched cards, punched paper tape, magnetic tape, parallel printer, typewriter, console).
Oct 1953	ATLAS II delivered; forerunner of commercial E.R.A. 1103 (UNIVAC Scientific 1103). Model 2, delivered in November 1954, equipped with core memory instead of electrostatic store; first core memory computer delivered to customer in U.S.
June 1957	LIGHTNING high-speed circuitry researches under way. Believed to be largest U.S. government computer research support. Influenced many commerical developments.
July 1957	First BOGART delivered. Believed to be first practical computer using magnetic (diode/core) logic in basic circuitry. Believed to be first computer to utilize design automation. Influenced design of several commercial models built by UNIVAC and Control Data Corp.
Mar 1958	SOLO delivered; first completely transistorized computer in U.S. Model for Philco's TRANSAC S-1000 and forerunner of improved S-2000.
Feb 1962	HARVEST delivered; most sophisticated model of STRETCH series. The TRACTOR tape system was the first completely automated tape library. Influenced design of IBM System 360.

features were later incorporated in such famed systems as the IBM 7090 and the 7000 Series.

Perhaps the greatest effort in computer cryptography saw its inception at a cocktail party in the middle of 1956, when NSA Director General Ralph Canine proposed a thousand megacycle machine to obtain an ultimate advantage over the flood of incoming data. Thus was born Project LIGHTNING. In the course of research and development, contributions were made by Sperry Rand's UNIVAC, RCA, IBM, Philco, General Electric, the Massachusetts Institute of Technology, Ohio State, and the University of Kansas. Untold spinoffs into the commercial sector issued forth, including over three hundred patent applications and seventy graduate theses. (The latest firm to be identified in the trade press with NSA is the Cray Corporation, said to have supplied a truly awesome-sized machine to meet the problems of the early 1980s.)

The Agency also aided the growth of integrated circuits, especially that indispensable essence of the computer industry, the large-scale integrated circuit. One gem, the "R-13" module, is supposed to have been supplied by most of the IC makers in the country at one time or another, despite the demands for great complexity and extreme reliability.

Beyond this, there is silence and the darkness of official secrecy, illuminated on occasion by brief announcements in electronic journals of NSA contract awards to companies which have systems with such arcane names as PARKHILL, VINSON, and ROMAN SHIELD. However, in due season, even the fruits of these highly classified, current endeavors will become known and appreciated. ▼

ROM'S ROBOT REVIEW PART III: WALKING THROUGH THE WOODS

by
A. I. Karshmer
and
J. M. Prager

Imagine you're a bird. You're away from home, but you know in what direction home is, and you want to get there. What do you do? Simple. You fly up in the air above all the trees and houses and just head for home. But now suppose you're an animal, and the place you want to get to is in the middle of the forest, and you can't fly. You may know the direction of your destination, but there are trees in your way, and your destination isn't in sight. Now, let's consider how we might program a robot to perform this task. We will place our robot at the edge of a forest, point him in the direction of his target, and set him moving. For simplicity, suppose that the target is something easily recognizable—say a bright light. We shall call our robot Jason, after the mythical hero whose task it was to find the Golden Fleece, hanging from a branch in the forest. But our Jason's task will be to reach the light without bumping into any trees.

The physical equipment required for any robot is a function of the task for which it has been designed. We can think of many uses for robots in the present or near future. Early robots will probably all be custom-built. They may, for example, have special robot arms for picking up and manipulating objects. We can envisage a future Viking landing and deploying a Mars-rover whose task is to scout around and pick up interesting looking rocks. It must be autonomous (that is, a robot), since direct control from Earth is impractical over such distances because radio signals take many minutes to complete the journey. Robots may be used also in all kinds of places where it would be too dangerous or costly to use humans—for example, mending underwater pipes, cables, and other deepsea equipment, or, back in space again, building a huge space station from materials brought up from Earth. Closer to our intended use of Jason, though, robots may be used instead of search parties in remote areas such as deserts, Arctic wastes, or even dense forests. All these robots will need the same basic equipment: a propulsion mechanism, a TV camera, and a minicomputer for guidance. Those that work on the ground (be it Earth or Mars!) might well consist of a flat board with wheels at the four corners, with the computer and camera mounted on top. The remaining space could then be used for power supplies, communications devices, and special-purpose equipment. (The robot sketched in figure 1 is similar to one now being developed at the Jet Propulsion Laboratory in Pasadena, California.)

Jason's task, then, may seem trivial—after all, we can walk through a wood without scarcely thinking about it, and so can animals. (Ever see a rabbit bump into a tree?) And our directions to Jason would seem to be straightforward: "If you come to a tree, go around it." But we couldn't really speak to Jason this way; instead, we would have to write a program for him which would do that for us. Actually, it's not that simple, you see, for how will Jason identify a tree? In other words, he has got to know that there is a tree standing in his way in order for him to go around it.

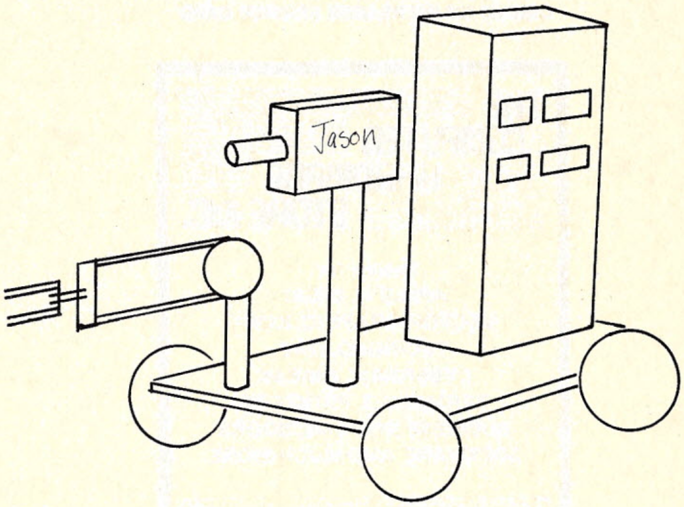


Figure 1
Jason, our robot

Why is that so difficult? Well, there are several problems. Suppose we were to embody a description of a tree in Jason's program, enabling him to compare what he sees with his internal description. But how do you describe a tree? One description, a brown cylinder, called the trunk, under a green spherical cluster of leaves, called the crown, might do for an oak; but firs and pines, for instance, don't have even nearly spherical crowns. A maple in the fall, for example, may be yellow or red, and it will have no leaves at all in the winter. A birch's bark is not brown, and some trees have foliage to the ground obscuring the trunk entirely. So describing a tree isn't all that straightforward.

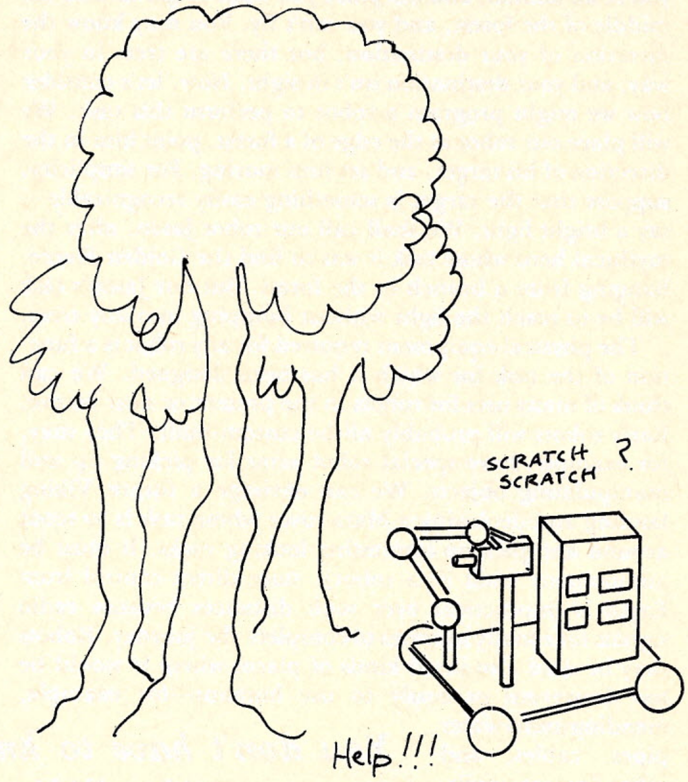
You don't have to know the species of a tree to avoid colliding with it, but you do need to know that it is there.

Therefore, in order to have Jason identify trees, he requires a great deal of semantic knowledge. Artificial-intelligence programs which do "image understanding," such as the VISIONS system previously described in this column, need large computers and take a long time—several minutes perhaps—to do their work. We can't afford this luxury,

especially if Jason is to be at all mobile. All we may allow him is a few seconds to perform his computations.

Although you don't have to know the species of a tree to avoid colliding with it, you do need to know that it is there. Maybe Jason can do some simple calculations on what he sees which will tell him this. In order to isolate the basic problems, let's make Jason's world a little simpler than the real one. First, let's suppose the ground is fairly flat, and there are no hidden roots, ditches, or holes lying-in-wait. We will also suppose that there are no overhanging branches at Jason's height, so all he has to contend with are the vertical tree trunks. To avoid collisions, Jason must be able to pick out each tree from the background and estimate its width and distance from him. He could use a laser range-finder to do this, but as this would be too hazardous if he is ever to "walk" among humans, we are insisting he use only visual information.

In a relatively, sparsely wooded area, it is easy to determine the sides of trees, since the trees will generally be dark against a brighter background (the sky), making the edges light-dark boundaries. But in dense forests, behind any tree you will find another tree; in fact, in all directions are trees. If from a given standpoint one tree is behind and slightly to the side of another, so that the first tree partially obscures or occludes the other, then how do you pick out the occluding edge of the front tree? This edge isn't a light-dark boundary as in a less dense forest—to one side is the



trunk of the nearer tree, to the other the trunk of the further tree. And these trees all might be approximately the same color. To make matters worse, the trees will probably have vertical markings and variations of shade (known as "texture") on the bark, making the edge even more difficult to pick out. In a sense, the trees camouflage each other. But we'll leave this problem for the moment.

Let's consider next the problem of how far away the trees are? Well, to solve this problem what we might do is copy

Nature and give Jason two "eyes," instead of one. The reason being that our two eyes give us what is known as binocular or stereoscopic vision, and with them we can tell the distance or "depth" of the things we see. Try covering up one eye: everything looks kind of flat. That's because each eye alone can only see that an object lies upon a certain line. In figure 2, the left eye knows that P is somewhere on line AB, and the right eye on line CD. When information from both eyes is combined, the intersection

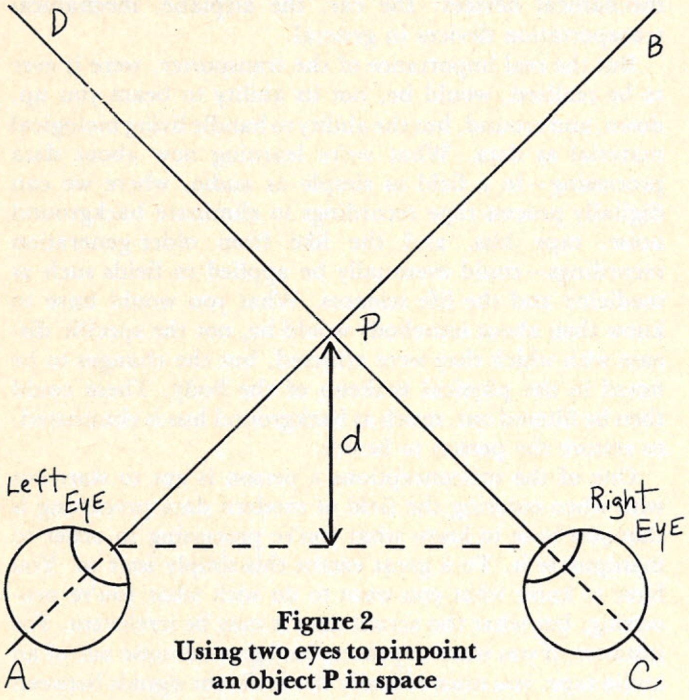


Figure 2
Using two eyes to pinpoint an object P in space

point P and the distance d can be easily calculated. The trouble is that our depth vision is really good only for distances of a few feet (d). For much larger distances, the lines AB and CD are nearly parallel, and the depth calculation gets very inaccurate. In order for Jason to have the same depth discrimination at one hundred feet as we have at two feet might require him to have his eyes (cameras) twelve to fifteen feet apart. He will be able to judge the gaps between the trees accurately, only now he won't be able to fit through them!

There are two more ways Jason might be able to get a handle on these seemingly elusive trees. First, the trees are not simply vertical trunks—they begin to branch out in all directions. By studying where the branches come from, Jason might be able to pinpoint the trunks exactly. Unfortunately, this also has its drawbacks, the biggest being that the computation will be excessive. (We may—unconsciously perhaps—use such techniques in our own maneuvers, but remember that animals with very small brains can run about in the woods—with more agility usually—and almost certainly they don't bother with the branches.)

The other trick Jason might use is to look for where the trees go into the ground. If he knows his own height and the angle his gaze makes with their vertical, then simple trigonometry will tell him how far away the tree is. (See figure 3.) This is a good method to use, but it runs into troubles if the ground isn't level, or if the point where the tree rises from the ground isn't clearly distinguishable.

So again we see that, if Jason is to use any of these methods, he will need a huge program which can cope with all kinds of strange exceptional cases that arise. This isn't very desirable, since Jason has other things to do. If he is an exploratory robot, he will be taking measurements of the local conditions. He might have special manipulative skills which he will use when he reaches his goal. Presumably, he must keep track of where he is going and maintain communication channels with his controller (which might be another computer). If he is to do all this efficiently, he can't have a large time-consuming program controlling his movement.

So far, we have been trying to help Jason determine his course from a stationary position. Supposing he just starts moving—what will happen? The answer is that now he can see much better! As you know, when you ride in a car and look out of the window, things by the roadside seem to just zip by, but buildings, trees, and so on, in the middle distance move more slowly, and the hills in the far distance don't seem to move at all. (Of course, none of these things are actually moving. Your motion relative to them causes them to appear to move relative to you—a phenomenon known as "induced movement.") You can even do an experiment right now—just move your head from side-to-side or back-to-front. Objects near you seem to move more quickly than those further away; the rate of induced movement is inversely proportional to distance.

So, this is how we can get Jason to calculate the distance of points in his field of view. It is a very simple relation based upon how quickly they appear to move when he

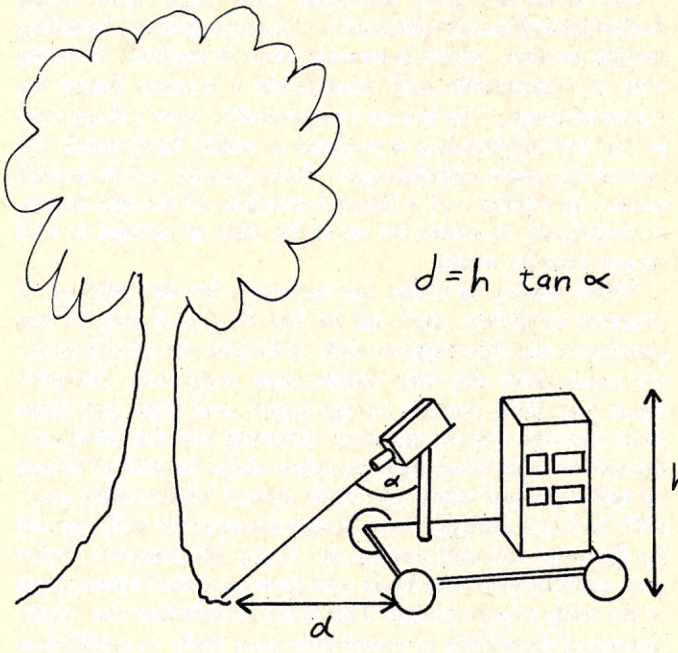


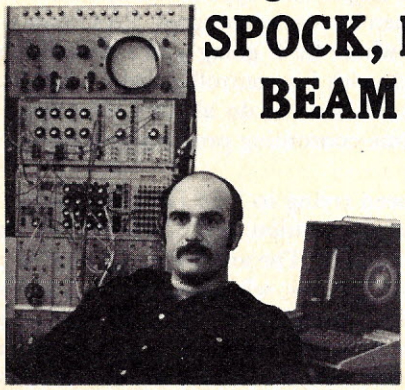
Figure 3
Determining the distance to a tree by trigonometry

does. Notice that we have also solved the problem of how wide the trees are. Recall the problem of one tree being in front of another. The points on the nearer tree will appear to move faster than those on the other tree, thus clearly identifying the boundary and grouping all the points of the individual trees together due to their similar motion.

Now that Jason can get along without bumping into things, the following question remains: how does he plot his path in an obstacle-ridden environment to reach his goal? But that will be discussed in another column. ▼

futuROMa

ONE QUESTION, MR. SPOCK, BEFORE I BEAM ABOARD



by Bill Etra

Remember the transporters in "Star Trek?" Here is a hypothetical device in which people stand under a to-us-undefined beam and have their atoms disassembled, then transported and reassembled at another location. We know this disassembly and reassembly is supposed to occur on the atomic level from hints given us during the show. At one point, for instance, Dr. McCoy says, "I signed aboard this starship to practice medicine, not to have my atoms scattered and reassembled all over the galaxy."

Fine as far as it goes, but what "Star Trek" never really deals with is the implication of this fantastic device: *handling people as data,* which is exactly what is implicit in being able to disassemble and reassemble a human being for transportation. The reason this probably wasn't dealt with is that the implication is so great it would have made the rest of the show unbelievable. Even fantasy needs a solid anchor in reality and a limited number of far-out facets, well-defined, in order for us to be able to accept it and escape into its world.

What reason, is there, for instance, for Dr. McCoy to practice medicine when all he has to do, if any of the people on the ship become sick, is to take the stored memory tapes from the time before they were sick, compare them with their current energy tapes, and then put them both through the transporter, filtering out the newly acquired viruses? What need to think about immortal beings on other planets when you have virtual immortality yourself? You can always return to the transporter and, merely by filtering out and saving the brain information, leave your current worn-out body and have yourself reassembled in the body of your youth. This is not to mention the implications of the ability to reassemble your body in a different form—or to borrow someone else's. More muscular, less muscular—think of the implications for people desiring a sex change. Take another space show, "Space 1999." In the second set of the series there's a woman who can reassemble herself into various animals. That would also be possible, obviously, if you could keep the brain information separate. You might run into problems with physical storage in animal-sized brains. But you could at least transmit or transform part of the personality of a human being into any animal for which you had the information tape stored. On one level, the transporter—though I'm not implying we have any way of implementing this type of device—is the electronic answer to replacing the last of the great mechanical devices: the car, the airplane, mechanical transportation devices in general.

But the real importance of the transporter, were it ever to be realized, would be, not its ability to beam you up, down, and around, but the ability to handle living biological material as data. What we're learning now about data processing—in a field as simple as audio, where we can digitally process tape recordings to eliminate background noise, tape hiss, and the like from older-generation recordings—could eventually be applied to fields such as medicine and the life sciences. What you would have to know then about somebody would be, not the specific disease with which they were afflicted, but the changes to be noted in the physical makeup of the body. These could then be filtered out, much as background hiss is eliminated, to restore the person to health.

One of the misconceptions a person is apt to start out with when entering the field of modern data processing is that you have to know what you're processing in order to manipulate it. To a great extent this simply isn't so. You have to know what you want to do with what you're processing, but what the actual data is may be irrelevant. For instance, if you want to process background noise out of an audio tape, you must identify the different signals between a clean portion and a noisy portion of the tape. But you certainly don't have to know what all the instruments making the music are doing. In another instance, you might want to know what the flute is doing because you want to know the range of the flute and you want to clean

CRYPTOGRAM SOLUTION

"THE NATIONAL SECURITY AGENCY IS THE OFFSPRING OF A HOST OF INDEPENDENT AND OFTEN COMPETING ARMY NAVY AND OTHER GOVERNMENTAL ORGANIZATIONS DESIGNING WITH CRYPTO SYSTEMS." (E = space)

Solution to last month's PROMpuzzle

E	V	E	R		S	I	G	N		F	E	T		E	R	S
W	I	R	E		T	R	U	E		A	L	A		D	I	E
E	T	A	S		R	O	S	E		R	I	B		I	C	E
R	E	S	O	N	A	N	T		P	A	T		A	T	O	D
			L	E	Y	S		M	O	D	E	L	S			
D	R	I	V	E	S		S	O	L			A	I	D	E	S
O	A	R	E	D		C	O	D	E	S		R	A	N	G	E
S	P	A	R		U	R	D	U		E	C	G		S	O	T
			S	E	R	I	A	L	A	D	D	E	R			
S	P	E		N	N	E		A	L	A	S		H	A	L	L
H	E	A	R	T		S	A	T	A	N		E	E	R	I	E
E	R	R	O	R			P	E	N		S	T	O	R	E	S
			B	Y	P	A	S	S		G	I	T	S			
A	G	E	S		O	D	E		H	O	L	E	T	R	A	P
L	I	T		P	R	E		M	A	R	I		A	I	M	E
A	G	O		C	T	P		A	L	E	C		T	O	M	E
R	O	N		S	S	T		E	T	N	A		S	T	O	P

the flute up for this specific range. But in that case you still don't have to know whether the music involved is Bach or Beethoven.

In the case of the transporter, if you couldn't separately identify brain material, then having yourself re-formed would take you back only to the moment and the memory and the time space in which you had recorded your information. Which would no doubt give you a sudden feeling of déjà vu déjà vu déjà vu much like a Monty Python sketch. On the other hand, if you could refine the process enough to know which part of the information was the brain, if you could take your current brain information and put it into your old body, then you would be involved in something much more complex—not to mention something inviting a lot of moral questions. If you can process a person, if you can transport that person's molecular structure and reassemble it somewhere else, then why can't you, by storing this data and adding the basic energy or atomic structure necessary, create a human being? Take so many pints of water and so many pounds of basic minerals, process them from a tape, and recreate a current valid copy? Cloning by data processing!

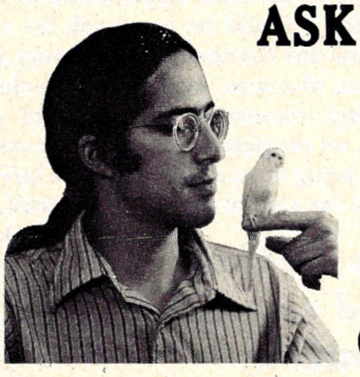
Handling people as data is implicit in being able to disassemble and reassemble a human being for transportation.

"Star Trek" does go into this possibility. There's an android-creation sequence in "Star Trek" where various members of the *Enterprise* are being duplicated. But it's done electromechanically. Why not merely take the transporter tape for reassemblage and feed in the required basic elements? (The footnote to this is a story by the science-fiction writer Larry Niven, in which Niven proposes that the transporter system be used for immortality. He doesn't go into some of the other applications of it, however, because, again, it would be too much for us as readers to accept within the framework of the story.)

At present, of course, we don't have a clue to how to disassemble and reassemble anything into another atomic form. Still, we do already know how to create new forms. In fact, it's being done all the time. It's how the last several elements on the atomic charts were created, artificially in a cyclotron. There is even some hope of creating solid matter out of energy, if you believe Einstein's theory of energy equals mass times the speed of light squared. You can fiddle with the equation and get mass on the other side of the equal sign. But making one or two atoms of a specific element in a very large cyclotron is a far cry from assembling a human being in the same manner.

Even so, you might let your mind play awhile with the thought of man as data, with an information theory of morals, with the idea of processing emotions and filtering out pain. As farfetched as it all might seem now, consider the fact that when Huxley wrote *Brave New World,* he didn't even hypothesize atomic energy, much less data processing—and that was just yesterday. ▼

PROMqueries
HAVE A PROBLEM? ASK ROM



by Eben Ostby

Dear ROM,

What's the best computer language? Although I'm just getting started with computers, I think I know the answer already—there is no single, best language. If that's the case, is there a best language for a specific task? I would like to use my computer to automate a greenhouse and keep hybridization records.

Robert Stickney
Urbana, IL

Dear Robert,

As you suggest, there is no "best" computer language. Were there such a language, all the other computer languages would have disappeared, since they all have various shortcomings. For years, the most popular languages were COBOL, ALGOL, and FORTRAN. COBOL is a good language for business processing—it's easy to produce reports and do accounting and record keeping in COBOL. COBOL is also wordy, which means it's a pain to program in, but the programs are fairly easy to decipher once they're written. ALGOL is a highly structured language—programs are written as a series of blocks, one inside the other. It is a very flexible language, and quite readable, and so has become a standard language for communicating programs between people. For instance, most algorithms published in academic journals are written in some version of ALGOL. FORTRAN is one of the oldest and most popular languages and is very good for straight numerical computation. FORTRAN is widely used for scientific calculation.

When IBM set out to create a new language, they had the brilliant idea that a language which combined the best features of these three would make them all obsolete. So they created PL/I, which is a very complex language encompassing the structure of ALGOL, the computational power of FORTRAN, and the data-manipulation facilities of COBOL. Unfortunately, PL/I is so big that it's practically impossible to learn all of the language. And, as a result, PL/I has not replaced any of the three languages it was supposed to put out of business. (Probably PL/I wouldn't have lived to this day if it hadn't had the backing of the largest firm in the industry.) All this goes to show that it's pretty unlikely that there will ever be a "best" language. You'll just have to balance the requirements of your application against what's available for your computer system.

By far the most available language is BASIC, which you undoubtedly know comes in all flavors and colors—from Tiny BASIC, through garden-variety standard BASICs, up to huge and powerful EXTENDED BASICs with fancy disk I/O. BASIC's biggest plus is that everyone knows it, and every computer has a BASIC interpreter or compiler available. It's not the most flexible language around, though—it doesn't have a lot of structure that makes programs easy to write, read, and debug. It also lacks certain niceties for dealing with characters and unusual data types. And it usually lacks facilities for doing I/O with anything but the standard devices (printers, keyboards, mass storage). So BASIC would probably be out for automating your greenhouse, unless you can get a version that has these special facilities.

My favorite language is APL (A Programming Language), which has the backing of Mother IBM. APL is really easy to program in, since it does multiple operations at the stroke of a key, thus removing you one step from the drudgery of programming. In essence, APL lets you concentrate more on what the computer is to do and less on how it is to be done.

APL isn't too popular among the 8080 set, though, since it takes lots of memory to run—around 32K bytes. Although a few APL interpreters are beginning to appear on home machines, it's not yet widely available to those of us who can't afford an IBM 5110 (or a 370). I expect this to change, though.

Another reason APL is unpopular is because it's so different from other languages. It's a strange-looking language, and it doesn't mesh with current philosophy that programs should read like ungrammatical English. It doesn't have certain "normal" programming constructs, such as loops and if-statements. But in spite of all this, APL is being used more and more.

The language you'll probably have to use to automate your greenhouse is assembly language. Assembly language has the flexibility to do pretty much anything. The only drawback is that doing something in assembly language is usually more difficult than doing it in some other language.

Certain computers have particularly interesting languages available to them. There are variants on PL/I that can be run on some micros (PL/M, among others). The DEC PDP-11 has a lot of interesting software, including FOCAL, which is somewhat like BASIC but nicer, and "C," which is like ALGOL but a bit simpler. There are countless others—LISP, Snobol, SPSS, GPSS, Simula, CLU, RPG, SLIP, Pascal, and so on—each of which is really good at doing some things and really bad at doing others. Got the picture?

ROM

Dear ROM,

There seem to be all kinds of printers, from daisy-wheels, to bells, to ink jets, and so on. What are the advantages and the disadvantages of some of the different types of printers?

Edward Marsh
Lansing, MI

Dear Edward,

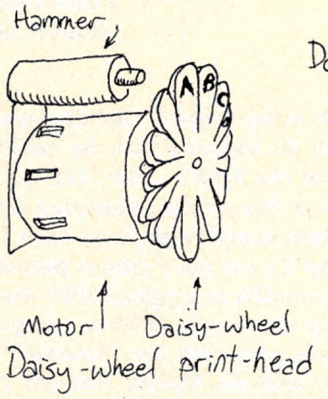
There are about as many different printers around as there are inventors who know about computers. In no par-

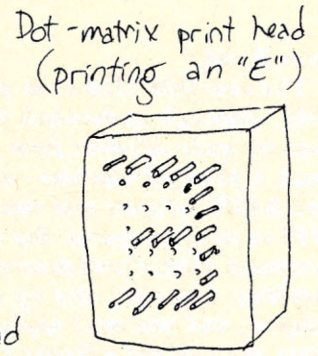
ticular order, here are some of the more common varieties.

The IBM Selectric typewriter and its cousins, the 2740, 2741, Trendata-1000, Datel 30, and many other Selectric-based terminals have the familiar noisy, complicated innards that are used in office typewriters around the world. They all use the handy replaceable type-ball. Although they produce beautiful output when adjusted correctly, they tend to break down when used heavily (just visit a big university computer center to see). They don't use ASCII code, for the most part; instead, they use IBM's EBCDIC and correspondence data codes. This means when you transmit an ASCII A to one of these, it prints something else. In general, they can be difficult to interface with an ordinary micro. But there are sure a lot of them around.

Then there's the old familiar teletype, models 33 and 35 (and its newer cousins). The teletype prints in a manner similar to the Selectric and other typewriters: it bangs little metal keys against the paper. Teletypes are cheap, clunky, reliable, noisy, and very common. If you're concerned about how things look, though, beware: teletype print looks like, well, like teletype print.

The daisy-wheel printers, like the Diablo (made by Xerox), are particularly nice devices. For one thing, they are fast—they can print some forty-five-characters-per-second (teletypes can manage about ten). The letters are very neat-looking, and the type-face can be changed almost as easily as that of the Selectric. They are also simple to use and quiet. In fact, they have just four basic moving parts: a daisy-wheel, a carriage, a platen, and a little hammer. The daisy-wheel spins around until the letter that you want printed is at the top of the wheel—each petal of the daisy has just one letter embossed on it. At the instant the letter is in position, the little hammer strikes it, printing the letter. Then the carriage (actually the thing that holds the daisy-wheel and the hammer) moves one space to the right, and the process starts again. Since it can move while the daisy is spinning around to the next character, it works quite fast. And since the tricky stuff—like determining how far to move the daisy-wheel or the carriage—is done electronically in logic, as are all of the important timing delays, the machine is quiet, fast, and even somewhat reliable.





Wire printers or dot-matrix printers are even faster than the daisy-wheel printers, but the print doesn't look as nice. They work by passing a series of tiny "hammers" across the page. Each time the printer is to print a letter, the right hammers strike the page. There are about thirty-five of these little hammers, arranged in a five-by-seven rectangle. If you use just the hammers along the left side, the five along the top, the five along the bottom, and the left-hand four in the middle, you can print an E.

Dot-matrix printers are quiet, but they do make a zizzing noise that sounds like a noisy zipper as they print each line. They are also pretty common, with everyone from Digital Equipment to IBM to Centronics making them.

There are a whole bunch of printers that are similar to dot-matrix printers in concept. But most use different methods of actually getting the letter onto the surface of the paper. Some of the more ingenious include spray printers, which, in effect, replace the little hammers with tiny sprayguns of ink that can be turned on or off at will; electrostatic printers, which use electrically live wires instead of hammers, printing on paper that is sensitive to the current; and thermal printers (found on some calculators), which use little heated wires instead of hammers and print on heat-sensitive paper which has to be ordered from California.

Then there are big *printers—line printers. A line printer somehow manages to accept an entire line of output at one gulp and prints it almost in one blow. They have a chain, with all the possible letters on it, which spins around in front of the paper. There are 132 hammers behind the chain, which hit the appropriate letters as they go by. By printing many characters at the same time, line printers can go really fast, producing thousands of lines every minute.*

Perhaps the ultimate in printers is waiting for you, if you have an IBM 370 that you can connect it to. This wondrous device is a laser printer, which uses a laser beam to burn the print onto the surface of the paper. One drawback of this incredible machine is that it can't make carbon copies. But then again, who needs carbon copies when this thing can spew out four pages in the same length of time its predecessor took to print one page?

ROM

Dear ROM,

Can you tell me how I might display some sort of animated graphics on a TV monitor connected to my micro? Are three-dimensional graphics possible? How about a rotating cube?

Sarah Goodale
White Plains, NY

Dear Sarah,

Anything is possible, but depending on your computer it may be a more or less convincing display. Graphics take a lot of computation to carry out properly. The basic elements you need are a program that will draw a line on whatever kind of display you have and in whatever position you specify, and a program to multiply the matrices together.

To get started, store the coordinates of the ends of the lines you want to draw in a matrix which has three rows (since we live in a three-dimensional world). To rotate the thing θ degrees around, say, the z axis, multiply each column of the matrix by the following matrix:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The result of this will have three rows, too, and as many columns as the original matrix of points. Finally, you display the x- and y-coordinates of the result on the screen (as

lines connecting the points). To make the thing turn convincingly, you've got to update the image frequently—at least twenty times each second. And that's the real problem—matrix multiplication takes time, and you may not have enough time to do the manipulation often enough to keep the movement smooth.

If you don't know what matrix multiplication is, it's an operation involving lots of multiplications and additions. For example, if you have a matrix A with M rows and N columns and a matrix B with N rows and P columns, this little BASIC program will multiply them together and put the result in C. (In APL, the program can be written C ←A+.×B.)

```
7001 FOR I=1 TO P
7005 FOR J=1 TO M
7010 C(J,I)=0
7015 FOR K=1 TO N
7020 C(J,I)=C(J,I)+A(J,K)*B(K,I)
7025 NEXT K
7030 NEXT J
7035 NEXT I
.
.
.
```

Yes, there is a lot of work to graphics—drawing things with your computer can be very hard. But it's also one of the most interesting things you can have your computer do.

ROM

Dear ROM,

While I was reading about computer languages, I came across the term "Backus Normal Form" without any real explanation for it. Could you please give me one?

Ray Branch
Boulder, CO

Dear Ray,

Backus Normal Form, also known as Backus-Naur Form or BNF, is a "language" used for specifying what other languages are. In particular, it's often used for describing programming languages. It was developed by Backus and first used for describing the language, ALGOL '60. Since the man responsible for finding this use was Naur, his name is sometimes included in the name of the "language."

Before BNF came along, programming languages were described pretty much in English. For instance, a DO-loop looks like this:

DO I≠ L TO N BY S, where I is any variable, and L, N, and S are any integers. . . .

BNF is a very precise way of specifying how a programming language is written. A BNF definition is a number of lines with a left-hand part, which specifies the name of the thing you're describing, and a right-hand part, which specifies what the choices are. For example:

⟨digit⟩ : : = 1|2|3|4|5|6|7|8|9|0

This says that a "digit" is either a 1, or a 2, or a 3. . . . The vertical bars indicate a choice between different options. Here, there are ten different choices: pick one.

In the following example, an "integer" is either a single digit, or else it is a single digit followed by an integer. Thus:

⟨integer⟩ : : = ⟨digit⟩ | ⟨digit⟩ ⟨integer⟩

Now, it may not make a lot of sense to you to define something in terms of itself, but it really isn't a meaningless definition. Since an integer here could be a single digit, you could replace the single part of the definition

⟨digit⟩ ⟨integer⟩

with the definition

⟨digit⟩ ⟨digit⟩

But since this, too, is an "integer," you could replace the above definitions with three digits. Before you ditch it, I'll explain more clearly: the definition says that an integer is just a string of one or more digits.

To befuddle you completely, I'll list a few more definitions. With them, you should be able to write any number that BASIC allows.

⟨signed no.⟩ : : = ⟨integer⟩ | + ⟨integer⟩ | — ⟨integer⟩
⟨fixed number⟩ : : = ⟨signed no.⟩ | ⟨signed no.⟩ . | ⟨signed no.⟩ . ⟨integer⟩
⟨E-part⟩ : : = E ⟨signed no.⟩
⟨real number⟩ : : = ⟨fixed number⟩ | ⟨fixed number⟩ ⟨E-part⟩

You've got to admit, it's precise.

ROM

Dear ROM,

Do you think the IBM 5110 could be considered a real personal computer? Would you recommend it for home video games and educational purposes? Or would I be better off buying something less expensive and concentrating the extra cash on peripherals?

David Porter
Boston, MA

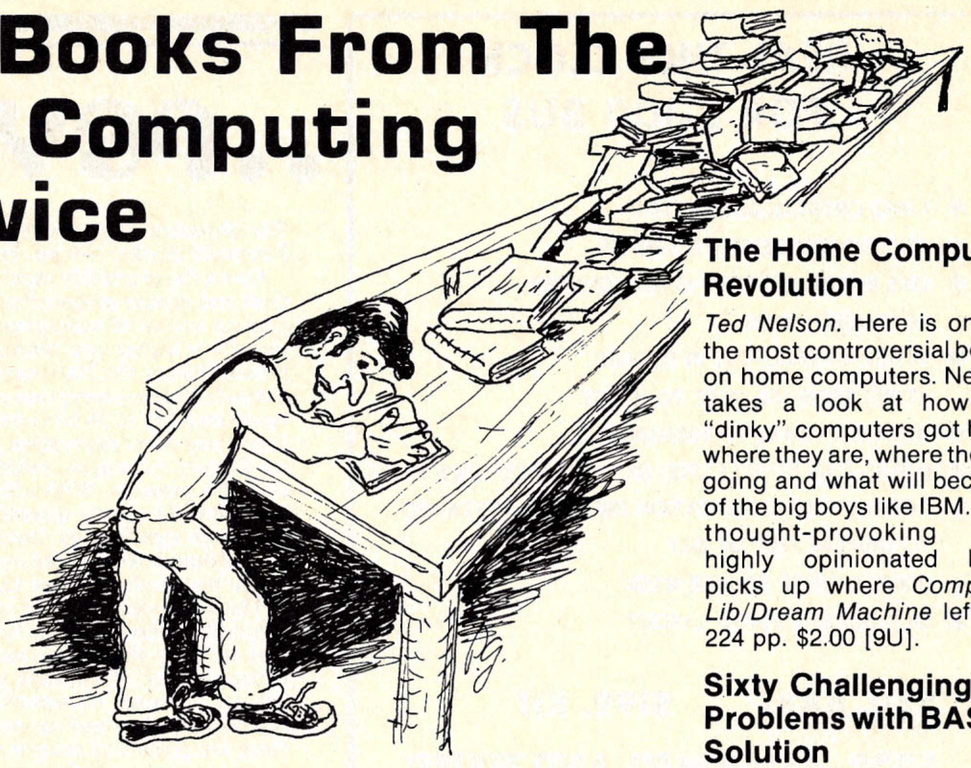
Dear David,

The new IBM 5110 is, in many respects, an up-graded 5100, although underneath its textured skin lie innards that are quite different from the 5100's. The 5110 is, indeed, a powerful machine—it has very sophisticated APL and BASIC software and runs many times faster than the 5100 on some programs. But it's not your typical personal computer. First, it's not as accessible as a typical 8080-based machine. You can't wire up accessories without using an adapter, and you can't easily play with the "internals"—those parts of the software that lie hidden beneath the elaborate systems operations that users normally get to use. You can't, for instance, find out exactly what's on a tape; nor can you do fancy tricks with I/O devices. So it would be a poor—and expensive—machine to buy for video games.

If you are interested in doing many business-oriented applications or scientific data processing, and no systems work, the 5110 is a fine machine—if you can afford it.

ROM

10 New Books From The Creative Computing Bookservice



The Little Book of BASIC Style: How To Write a Program You Can Read

John M. Nevison. Learn how to write better, easy-to-follow programs with Nevison's rules of style and turn out legible, correct programs. Two hours of BASIC programming is all that is necessary to profit by this book. Concepts of problem-solving and structured programming are included. 160 pp. $5.95 [9V].

My Computer Likes Me

This entertaining self-teaching workbook introduces the BASIC language to young or old. Problems and examples are drawn from population problems and demographic data. A nice, easy start into BASIC. Large format. 60 pp. $2.00 [8K].

BASIC and the Personal Computer

Dwyer and *Critchfield.* This book will get you involved with personal computing, writing programs and expanding the use of your computer by showing the great diversity of applications possible on any microcomputer. One of the most comprehensive presentations of BASIC ever. As a text or addition to your personal library, this book will tell you all you ever wanted to know about BASIC. 350 pp. $12.95 [9F].

Games With The Pocket Calculator

Sivasailam Thiagarajan and Harold Stolovitch. A big step beyond tricks and puzzles with the hand calculator, the two dozen games of chance and strategy in this clever new book involve two or more players in conflict and competition. A single inexpensive four-banger is all you need to play. Large format. 55 pp. $2.00 [8H].

Beginner's Guide To Microprocessors

Charles M. Gilmore. No background in electronics is necessary to understand this book. It was written for those with no prior knowledge whatsoever of microprocessors or personal computing. Gilmore takes you from what a microprocessor is, how it works and what it's used for to how they're programmed to perform desired functions in microwave ovens, TV games, calculators, etc. 175 pp. $5.95 [7U].

Modern Digital Communications

E.J. Ross. This volume thoroughly explains how mass communications systems used by banks, businesses, airlines, news services, hotels and others operate. A must for anyone interested in computerized communications, including computer hobbyists and amateur radio operators. 304 pp. $6.95 [7V].

Problems For Computer Solution

Steve Rogowski. The Student Edition is designed to encourage research and preliminary investigation on the part of the student. The problems are ordered by subject and can be expanded or shortened. Mathematical problems that have never been solved are also posed to challenge and sharpen the student's awareness. 98 pp. $3.95 [9Z]. Also available is the Teacher's Edition which contains solutions, programs and analysis of the problems. 271 pp. $9.95 [9Y]. Both books are highly recommended for any high school or college computer-oriented course.

The Home Computer Revolution

Ted Nelson. Here is one of the most controversial books on home computers. Nelson takes a look at how the "dinky" computers got here, where they are, where they're going and what will become of the big boys like IBM. This thought-provoking and highly opinionated book picks up where *Computer Lib/Dream Machine* left off. 224 pp. $2.00 [9U].

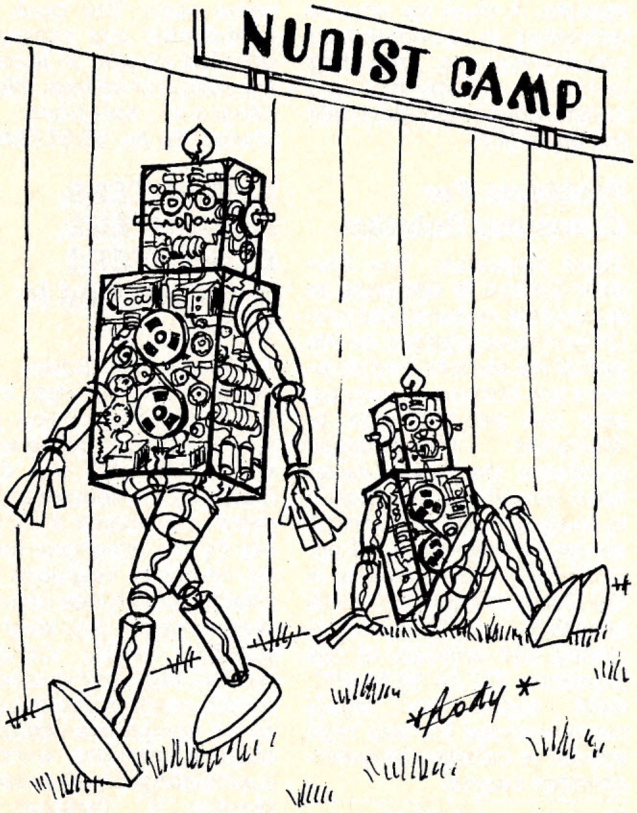
Sixty Challenging Problems with BASIC Solution

Donald Spencer. This book is a vehicle for computer programmers to measure their skills against some interesting problems that lend themselves to computer solution. It includes games, puzzles, mathematical recreations and science and business problems — some hard, some easy. The book will compliment any computer-oriented course in secondary school or college. BASIC program solutions included. 80 pp. $6.95 [9W].

COMPUTERS, COMPUTERS, COMPUTERS In Fiction And In Verse

D. Van Tassel, Editor. This collection of stories, commentaries and poems project the reader into a world where lifestyles are dominated by the computer to an extent far greater than they are by the telephone today. By revealing reactions and effects, the stories offer the reader insight into what is a potential reality. Cleverly-written, this book should entertain anyone who is aware of the computer's impact on society. Includes work by such distinguished writers as Gordon R. Dickson, Art Buchwald, Michael Shaara and Bob and Ray. 192 pp. $6.95 [9X].

To order call toll-free
800-631-8112
(in NJ call 201-540-0445)
fill in the inserted order card or write to:
creative computing, Attn: Marie, P.O. Box 789-M, Morristown, NJ 07960.
*All book orders must be prepaid. Include $1 for shipping, USA; $2, foreign.



..reviews.

The Standard Data Encryption Algorithm. Harry Katzan, Jr. Petrocelli Books. 144 pp. Hardbound. $12.00. 1978.

Given the incredibly rapid rise of computer crime through the theft and misuse of stored or transmitted data, coupled with the coming advent of widespread EFT networks, the need for data security is higher now than ever before. After much public and private debate, the National Bureau of Standards has developed a standard data-encryption algorithm. The algorithm selected is based on a key of 56 bits and is structured to eliminate repetitive letters and other decryption giveaways. This means that there is no known technique, other than trying all possible keys to decode a message. With a 56-bit key there are over 70,000,000,-000,000,000 (seventy quadrillion) possible combinations. On a computer significantly faster than any that exists today, this type of search would take over 2500 years to derive the key even if you had both the clear *and* encrypted text of a message.

The book describes the standard algorithm in detail in the second half; however, the first part of the book is fascinating in that it takes the reader on a very understandable and non-technical tour of virtually all encryption methods. Professor Katzan discusses transposition ciphers, substitution systems, algebraic systems and various combinations. Examples of all of them are given and several are further "illustrated" with APL programs for enciphering and deciphering. Recommended reading.

David H. Ahl

*****

Computers and Their Societal Impact. Martin O. Holoien. John Wiley & Sons, N.Y. 264 pp., hardbound. $10.95. 1977.

This book starts out covering standard historical developments that led to the invention of the modern computers. Despite the tired ground that is covered, the author gives the cast of characters a new twist by supplying mini-biographies of each, even including the ill-fated Lady Lovelace. The assumption seems to be if you tell the reader more about the humans involved, the computer will seem less awesome.

The easy conversational style of writing begun in these first two chapters is carried out throughout the book. As a matter of fact, the book carries this tone so far that I wondered whether I was reading printout from ELIZA (the psychiatrist program), or from a teaching program which inserts the appropriate mix of *we, you realize* and *you know*'s into the text. This gets overdone to the point where the author seems to be talking down to the reader, probably because of the intended audience for the book.

The book was written on the premise that every adult should know something about computers and that each person should find out about the potential benefits that can result from knowing how to use a computer. The book is intended as a text for 11th and 12th graders and college freshmen, and was probably constructed from the author's class notes and tapes.

The book is organized for this purpose. Each chapter has readable text followed by exercises and some well-chosen references. The exercises are broadly constructed, some requiring short answers, others suitable for short research or term papers. Most are appropriate to the high-school level or for non-technical students. Enough pictures and diagrams are included to enhance, but not clutter the text. I appreciated the completeness of the programming examples included. The author uses complete programs or logical parts of programs. Many other books give the reader the dummy treatment, showing only a measured dose of code unrelated to any whole process, presumably because that is all that can be absorbed.

There are two useful appendices geared to the beginning student, one explaining flowcharting as a problem-solving technique, and the other explaining how to use a terminal.

The majority of the book is a survey of computer applications in education, business and manufacturing, politics and government, law enforcement, and health, complete with some good examples. Many of the applications are handled concretely, with a specific example of a useful program at the

code level, and what it does for the manager, policeman, politician, or other user. These build on an exceptionally well-developed chapter that teaches the reader enough BASIC to be able to program and follow the examples. The obligatory future chapter which ends the book is uninspired.

Overall, I'd recommend this book, but not to the regular reader of *Creative Computing*. Give it to your Congressman, your minister, your maiden aunt, or to non-science or math majors with whom you'd like to have common ground. It could be a good text for students with non-technical orientations. It's awfully lightweight for anyone who has already been exposed to computers.

Deanna J. Dragunas
Wetumpka, AL

*****

Artificial Intelligence. Patrick H. Winston. Addison-Wesley Publishing Company. 444 pp. hardbound. $14.95. 1977.

Winston's book isalmost certainly the best seriousintroduction to the subject available today and, given the limitations of the field, it is very good indeed. Anyone who has tried to teach a course on Artificial Intelligence will recognize the difficulties that Winston faced in trying to organize a textbook on the subject. The field is unusual in that most of its primary results are simply Ph.D. theses in which some intelligent student (often at M.I.T., where Winston heads the Artificial Intelligence Laboratory) has written a clever program that seems to behave intelligently. As a result, the field can give the appearance of being just one clever program after the other. Winston comes close to avoiding this appearance by typing results together in interesting ways and showing a unity of technique that is beginning to become apparent in the field.

The book is divided into two parts. The first part, consisting of 9 chapters, discusses the key ideas of Artificial Intelligence (or AI). The second part, consisting of 7 chapters, is an introduction to LISP. Because these latter chapters are tied directly to the material in the first part, the reader of this book has, as Winston points out in his Preface, several options. The reader can just go after the key ideas, focsing on the material in the first part. Or he or she can just go after the second part which contains about as good an introduction to LISP as I have seen, Or one might pursue both parts in parallel, using the programming part with LISP to provide serious substance to the more generally presented materials in the first part. And there is a fourth way of going after this material that readers of *Creative Computing* might prefer. One might write the kinds of programs discussed in Winston's first part in some other language (say BASIC) thus providing more substance to the material in that part without having to learn LISP. There is something enormously informative about trying to write specific programs to do what a general description of a program says it is going to do. Writing such programs is probably the only way that one can really appreciate the difficulties and learn what the basic ideas really are.

But no matter how one approaches the subject, one is probably going to find this book both good and hard. It is not hard because Winston is unclear (he is very clear indeed) but because the underlying ideas are often quite difficult.

The book begins with a motivating chapter about "The Intelligent Computer" in which Winston describes the nature of Artificial Intelligence and the nature of *Artificial Intelligence* (which is to say, of both the subject and of this particular book about the subject). He argues that intelligent computers could have practical benefits and that studying intelligent computer programs can give us some insights into our own intelligence. He then proceeds to show us some examples of the kinds of intelligent things that computers can do and that will be discussed in more detail in the rest of the book.

The book ends with some "problems to think about" that provide exercises for the first part that can be done by people who don't know how to program. This is a very useful feature of the book.

This book is probably the best introduction to the field of Artificial Intelligence for a reader of *Creative Computing* who is willing to work hard and also has the fairly high level of intelligence needed to understand this book without the help of a teacher of some kind.

Peter Kugel
Chestnut Hill, MA

Computers and the Cybernetic Society. Michael A. Arbib. Academic Press. 494 pp., hardbound. $12.95. 1977.

Books about computers and society are a bit like mothers and apple pies. The underlying concept is appealing but the reality is often disappointing. Apple pies burn, or their crusts get soggy. Mothers are not all they are cracked up to be. And books about computers and society can be disappointing to even those of us who are in favor of both computers and society.

But this book about computers and society from the prolific pen of Michael Arbib is better than most and, indeed, may be the best this genre has to offer. Here is a book by a man who knows what he is talking about and who has tried to organize the material around basic and important ideas. This is not a technical book by any means but it does have some hard ideas in it that the student will have to think about.

The purpose of the book is well summarized by Arbib in his introduction "To the Student" as follows:

"This book was written to explain the impact the computer has on you today, and to provide the knowledge you need to help decide how computers will be used in the future. For this, you need to know what computers are and how they are programmed — how they can be given instructions to make them perform the jobs described above. Once you have this basic understanding, you can get involved in the many applications of computers, and form your own opinion of where we need greater use of computers, or where their use needs to be restricted or controlled."

The book begins with a chapter that introduces the idea of a program in a variety of interesting applications and in some rather unusual ways. Arbib begins by discussing a program to guide a robot around a room and immediately introduces the concept of a loop. This is a rather unusual way for a book of this sort to start but it makes a lot of sense. The idea of a loop is probably one of the more important ideas in computer programming and it is one of the things that beginning students, in my experience, seem to have the most difficulty with. Heading right to the heart of the matter may help.

Arbib then discusses the idea of a program in a somewhat different application — a digital watch — an application with more pedagogical value than it might, at first, appear to have. Arbib's ability to organize material so that it makes sense and fits together is illustrated by what he discusses next, the often made claim that computers can only do what they are told to do. He says this claim is wrong and there are two important reasons why this is so. The first is that computers don't always do what we think we are telling them to do, which gives Arbib the chance to discuss bugs. And the second is that programs can learn, which gives him a chance to discuss learning programs. Thus, rather than discussing these two important ideas separately, he ties them together under a single theme. This ability to tie diverse ideas together to underlying concepts is demonstrated again and again in this book and it is one of its greatest merits.

After the first 90 pages, at the end of Chapter 1, the diligent student should have a fairly good understanding of some important ideas in programming — including the ideas of bugs, loops, and recursion and also such ideas as the idea of a subroutine and of information structures.

These are deep ideas but Arbib's presentation of them is clear and interesting. I do not think that the average novice cold pick them up from the explanations in the book alone. But the person who knew something about computers already could read this part of the book with pleasure and, I think, with the result of increasing his or her understanding of the subjects covered.

Chapter 2 introduces some basic ideas about computers that Arbib (correctly) thinks to be important for an understanding of the role of computers in society: the ideas of compilers, operating systems, time sharing, graphics, networking, microcomputers and even of structured programming and program verification.

There follow 6 chapters on varios topics that could, as Arbib suggests, be studied in virtually any order one chooses. Chapter 3 deals with simulations. It begins with a general discussion of what simulations are, followed by a Prey/Predators example. This example is well worked out and fits together into a nice package. This is followed by a discussion of "The Limits of

Growth" that makes more sense to me than most of the things that have been written about it.

Chapter 4 discusses data banks. It begins by describing the problems involved in data base management, using an airlines reservation system as an example. After the basic ideas involved in data base management are introduced, Arbib proceeds to discuss EFTS, the "checkless society" and computer crime. Then he goes on to a discussion of the role of concept of privacy in a data bank.

Chapter 5 discusses Artificial Intelligence. In the style of the followers of Warren McCullough, Arbib begins, as many other workers in Artificial Intelligence probably would not, with the human brain. Then he discusses "machines that see and plan," combining a discussion of scene analysis with a discussion of planning. There is a section on "machines that understand natural language" and their use of the idea of microworlds which limits the domains of discourse, thus making the understanding problem simpler and more tractable. And finally this chapter has a section on "computers in war and peace" in which various military applications are discussed together with the moral issues such applications can raise.

Chapter 6 is titled "Learning and Working." It includes a discussion of the role of computers in education with good sections on the use of the computer for drill and grill, for tutorials in the style of PLATO and in what he calls the "Dialog-Inquiry" mode in the style of the late Jaime Carbonell (an approach whose merits totally escape me). There is a brief discussion of "the computer as an aid to discovery," as in the kind of thing they do in the SOLOWORKS.

The second section of this chapter deals with information flow, discussing the potential roles of computers in generating newspapers, in communications and in libraries. And there is a section on the impact of automation.

Chapter 7, on "Networks and Politics" discusses the potential role of computers in planning, their potential impact on democracies and their possible roles in worldwide planning.

Chapter 8 ties together various ideas about hardware and compiling. Here is another example of how Arbib ties together things that seem relatively far apart. How he does it is suggested by the title of the chapter "Down and Up from Machine Language."

Although this book is primarily intended as a text for a "Computer and Society" course, readers of *Creative Computing* might find it worth reading for fun. I don't believe that somebody who knows nothing about computers is going to get much out of this book by reading it by themselves. But somebody who already knows something about computers, will find an intelligent discussion of important ideas and issues here. The discussions vary in quality, but, by and large the level and the quality of the presentation are high. The typography and illustrations are good, but don't judge this book by its cover. Publishers seem to use some sort of random cover generator to produce covers for books on computers and society. The insides of this book, however, are probably as good an introduction to the subject of computers and society as you are going to get anywhere today.

Peter Kugel
Chestnut Hill, MA

*****

The First Book of Kim, reviewed in May-June 1978 (p 79) as being available from the authors, is published by Hayden Book Co., Rochelle Park, NJ 07662.

puzzles & problems

Want To Be a Millionaire?

David H. Ahl

I recently received the letter shown to the right giving me a unique opportunity to become a millionaire virtually overnight. I chose not to follow up on this fantastic opportunity not because it is in violation of the U.S. Postal Law (it is), but because a chain letter relies upon a geometric progression for its success. What's wrong with that? Simply that you very quickly use up the entire population of the U.S. and, indeed, the world before you "get rich."

Consider this letter which asks that each recipient send it to 500 other people (most chain letters settle for 10 other people).

Position on List	Total People in Chain
1	500
2	250,000
3	125,000,000
4	62,500,000,000
5	31,250,000,000,000
6	15,625,000,000,000,000

To continue would be ridiculous; however, bear in mind that to make any money by starting as the fourth name down (with a list of three names), your payoff comes from people at the sixth level. They would have to number some four million times the total population of the world for you to make your million.

Chain letters that ask you to send copies on to only ten people somehow sound more plausible, but are they? Work it out for yourself and see whether you can reasonably expect a payoff. Hint: there are about 70 million households in the USA.

$$$$$$$$$$$$$ MILLIONAIRE'S NEWSLETTER $$$$$$$$$$$$$

DEAR MILLIONAIRE: $$

We know this is an unusual way to open this issue, ie, calling those of you who may feel far from it---MILLIONAIRE'S NEWSLETTER is going to be an unusual edition of the MILLIONAIRE'S NEWSLETTER because one subsidiary in Bessemer, Alabama, has hit upon something BIG he is sharing with the rest of us that took him from his meager position of clerk in a small hole-in-the-wall store of a dinky little town. His small take home pay was $90.00 a week.

BUT at this time Steve Gilbert of that small town is reported to now be sitting of $737,988.00 all gain in less than three months by using this letter.

We have been experimenting with this letter ourselves and in just five weeks after sending out a mailing of only 500 pieces we have our own success to relate to you.

After just five weeks we are $17,898.00 stronger in revenue procured from only 500 pieces mailed. We wish now we had mailed 1,000 instead of 500. Our next mailing will be 5,000 and we may retire at the end of the year.

You have been HAND-PICKED for this OPPORTUNITY and we URGE you to read this letter carefully. Because this is SOLID and we so firmly know it and have our own proof of it that we're doing something we've never done before and that is stake our reputation on it. Steve Gilbert says he started with a mailing list of 500 and then quickly---followed it with 1000 quantity mailings as the MONEY began to pour in faster and faster.

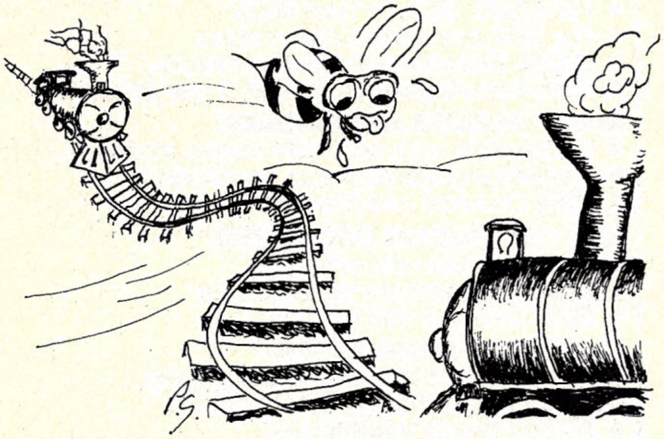
This is the most UNIQUE and original letter ever written and released to the public, so if you follow the forecoming procedures CAREFULLY you will soon be receiving CASH from 25,000 people who, like you, believe in helping others before they can expect others to help them.

Here is how you do it, Type the name list on a 3"x5" slip of paper OMITING the name in number one position and place your name and address in the THIRD position. MAIL $5.00 to the name in number one position if you don't, do not expect others to mail you your $5.00. Mail $5.00 as soon as you get this to #1 name. Paste the new list over the present list then photocopy it and take the new letter to your nearest printer who will copy them for 2 or 3 cents each. When your name reaches the #1 position, it will appear on enormously multiplied number of copies. At that point, you'll start receiving the $5.00 gifts. You need not worry about paying it back because your $5.00 sent to the number one name entitled you to MEMBERSHIP. Do not ALTER this LETTER. IT really is working for us and it will for you, ONLY if you use it right NOW. $$

If you do not have names and addresses of people to whom your letters can be mailed, you can send for Mailing List, Cost is about $38.50 per 1,000 names. Call or write: Addresses Unlimited, (213) 873-4114.
Shipping including – 14760 Oxnard St., Van Nuys, Cal.

****

NO. ONE-EDDIE DILL'S, Rt. No 1, Box 115-B, Blue Ridge, Ga. 30513
NO. TWO-G.L. GILLHAM, Rt. 1, Lot, 150, Franklin, Ga. 30217
NO. THREE* J. GOUGHNOUR, Box 389A, Mineral Point, Pa. 15942



Train and Bee

Two trains are converging on the same track. One is going 11 km./hr and the other is going 22 km./hr. A bee is flying from one to the other at a constant rate of 33 km./hr., back and forth, back and forth. How far will the bee fly in the last hour before the trains collide?

Sanderson M. Smith

Goops, Gorps, and Gorgs

Every GOOP is a GORP.
Half of all GORGS are GORPS.
Half of all GORPS are GOOPS.
There are 40 GORGS and 30 GOOPS.
No GORG is a GOOP.
How many GORPS are neither GOOPS nor GORGS?

Sanderson M. Smith

Square Root, Root, Root, Root

What is the value of:

$$\sqrt{12+\sqrt{12+\sqrt{12+\sqrt{12+\sqrt{12+\ldots}}}}}\ ?$$

Sanderson M. Smith
Chairman, Math Dept.
Cate School
Carpinteria, CA

(Answers on page 141)

business computing

Overview

WORD PROCESSING: The Office of the Future is With Us Today

John B. Hayes

Low-cost microcomputers and peripherals have brought the capability of sophisticated word processing and text editing within the reach of many new users in education, government, business, and the professions. A number of word-processing systems and software packages make the low-cost, general-purpose microcomputer a feasible alternative to a single-purpose word-processing system. However, the wide variety presents the potential new user with a number of critical considerations. This article provides guidance to what a word-processing system can and should do.

The term *word processing* describes the creation, editing, and printing of text material with considerable user control over formatting and related functions. The terms used in the industry today are somewhat confusing because a number of manufacturers include everything from dictating machines to copiers as "word processing" equipment. The term *text editing* may be also applicable to this subject but the term implies certain limitations. A "word-processing" system should do a number of things other than simply *editing* text.

A Long Way in a Short Time

We marvel at the tremendous growth of the computer industry in a very short period of time — the first commercial computers were installed only about 25 years ago. However, the metamorphosis of word processing has taken place in even a shorter period of time. Until the late 1960s, automated word processing was limited to "automatic typewriters" that punched holes in paper tapes or rolls to produce repetitive letters. The early auto-typers used a wide paper roll, similar to a player piano roll, with each column of the roll corresponding to a printable or control character. The Friden Flexowriter or other "flexy"-type automatic typewriter operated with a punched paper tape using five to seven channels in a fashion very similar to Teletype machines. Editing of the paper tapes involved cutting and splicing new sections of paper tape to make changes in existing text material. Some of the later machines used a dual paper-tape system to allow a repetitive letter to be coded on one tape (glued end-to-end for repetition) with a name and address file on the other tape. The paper-tape machines were frequently used to generate repetitive correspondence.

Low-cost microcomputers and peripherals have brought the capability of sophisticated word-processing and text-editing within the reach of many new users in education, government, business, and the professions.

IBM's MT/ST

In the mid 1960s, IBM introduced the stoic MT/ST machine (Magnetic Tape/Selectric Typewriter) which replaced the paper tape with a magnetic-tape cartridge. Operation of the MT/ST was similar to that of the paper-tape machine in that individual characters were coded on each line of the tape. The MT/ST machines were an improvement over the flexies in that corrections could be made by retyping a character over an incorrect character without the need for splicing the tape. In addition, quite a number of short documents could be encoded on a single cartridge. The MT/ST machines had in common with the flexy machines a number of disadvantages including the fact that each document had to be addressed sequentially, since they were on a single tape. The major disadvantage of the early MT/ST machines may have been the fact that the control codes for the machines were not visible and could not be printed on the typewriter; the MT/ST machine operated "blind" when it transferred control from one tape to another. For example, when the name and address were to be inserted in a letter, the machine would type whatever was positioned on the address tape rather than taking information from a particular field. For example, "Dear Atlanta, Georgia 30303:" was not an uncommon salutation on repetitive letters in a MT/ST word-processing shop. An experienced operator could generally tell what control codes were on a tape by the way the machine reacted, but in certain cases it was just a guess. Another

John B. Hayes, Alston, Miller & Gaines, 1200 Citizens & Southern National Bank Building, Atlanta GA 30303.

major disadvantage of the MT/ST machine was its limited ability to do editing. For example, a few characters could be changed in a document, but the insertion of new words or lines required the document to be copied to another tape and the extra material inserted while it was being copied.

Improvements over the MT/ST

In the late 1960s, a number of manufacturers introduced machines that resembled the MT/ST. Many of these machines were also dual-cassette machines but several of them contained improvements over the MT/ST. For example, the Wang 1200 word-processing machine provided the operator with the ability to print control codes in order to see exactly what was on a document. The majority of these machines were dedicated, single-purpose machines for performing word-processing. The use of computers to do word processing also developed during the late 1960s and early 1970s. Shared logic systems were frequently available for limited text editing. IBM provided a text-editing package for administrative documents on the System 360. The package used a Selectric terminal to produce excellent typed reports. Although these early systems on large-scale computers provided flexible text editing, they were expensive to use on a large-scale basis.

Mini-Based Word Processing

In the early 1970s, several companies began assembling minicomputer-based word-processing systems using video display and cassette storage. Some of these systems were "shared logic" and were really general-purpose computers that could be used for other applications. A few of the more expensive systems used disk storage. It was surprising, however, that the companies producing the single-purpose word-processing systems did not seem to grasp the notion of using a small, general-purpose computer to drive their systems. For example, in the early 1970s, Wang Laboratories produced an efficient single-purpose word-processing machine, the Wang 1200, while at the same time a separate division produced an excellent, low-cost general-purpose computer called the Wang 2200. Wang had apparently made little effort to combine the desirable features of the two separate product lines to produce a shared-logic word-processing system. In 1973, one customer of Wang insisted on doing word processing on the 2200 computer. The result was something of a kluge contraption, although it is still in use today. Wang provided the customer with a model 2200 computer with hard disk and CRT display. Since Wang did not make an upper- and lower-case CRT, the video display presented only a line of upper-case characters. If a particular character was to be upper-case in the output, the next line on the CRT contained an arrow underneath that character. Therefore, the otherwise efficient 24-line CRT could only produce 12 lines of useable text information. Since Wang had no typewriter-quality printer at that time for its computer, the system generated a cassette that could be read by the Wang 1200 word-processing machine. The cassette was transferred to the word-processing machine which essentially operated as a slave printer. Although this system appears grotesque by today's standards, it was an inexpensive way to achieve computerized word processing at the time.

A "word-processing" system should do a number of things other than simply *editing* text.

Flexible Diskettes

The advent of the flexible diskette in the mid 1970s made inexpensive, computerized word processing a reality. A number of systems houses configured machines that used single or dual floppies as a random-storage device for text documents. The ability to access a document file in a random fashion provided for greatly improved editing and formatting capabilities. The microcomputer has now provided a way of reducing the cost of the systems. A substantial part of the cost of a system today is in the cost of the peripheral devices needed to do word processing, rather than in the processor. Market demand and volume may reduce the cost of the printers and diskette drives and further reduce the cost.

Hardware Requirements

The hardware required for word processing on a microcomputer depends to a large degree on the level of sophistication of the word-processing system. A simple text editor can store a small amount of information in memory. Since one ASCII character can be stored in a single byte in the computer, a full page of average text or a medium length letter can be stored in 2K bytes of memory. The advantage of text-editing systems, however, is their ability to store text "off-line" in some form of permanent storage. A cassette-based text-editing system is satisfactory for storing lines or paragraphs of text that do not have to be rearranged. The disadvantage of the MT/ST and other cassette word-processing machines is the inability to insert data in the middle of a document

or rearrange the document without the necessity of transferring the document from one cassette to another. Cassette storage in a microcomputer system has the same disadvantage.

To have any degree of flexibility, a word-processing system should be based on a random-storage device, such as a floppy disk. A floppy-disk user will soon find immense advantages in having a second diskette in the system because it allows flexibility in manipulating sections of documents as well as the all important safety feature of being able to back-up diskettes. The number and size of documents that can be stored on floppies depends largely on how the documents are organized and the access method used to reach the document. For example, one system that uses a single-density, single-sided, full-size diskette can store about 240,000 characters of information (in addition to the overhead storage required). A micro-floppy on the same system can store over 80,000 characters, which provides more than 30 single-spaced pages loaded with text information. A major advantage of a floppy-disk system is the relative ease of using a number of diskettes to contain different segments of a single document or different document files that are to be merged together to produce a single document.

Printer

The printer in a word-processing system is probably the most critical item in determining whether the system produces truly useful output. A number of the microcomputer-based word-processing systems (as well as many of the single-purpose word-processing machines) use daisy-wheel printers produced either by Diablo (a Xerox subsidiary) or Qume. These printers are capable of printing up to 45 to 55 characters per second, which means that a single line of text is printed in about one second and a full page of text in less than a minute. In addition, these printers have a "slew rate" higher than the printing speed which means that they will skip spaces and lines faster than if they were printing. The result is that many letters and pages of text are actually typed in twenty to thirty seconds. The daisy-wheel printers produce a typewriter-quality impression, particularly if they are used with a carbon ribbon.

CRT Display

A number of the microcomputer-based word-processing systems use a CRT for visual display. Such a display makes editing of text somewhat easier and allows the user to more rapidly scan documents. Another approach that lowers the cost of the systems is

To have any degree of flexibility, a word-processing system should be based on a random-storage device, such as a floppy disk.

the use of a hard-copy terminal such as the Diablo 1620 or a Selectric typewriter terminal. Some of these hard-copy terminals use the daisy-wheel printing mechanism and provide a keyboard so that the CRT display device is not required. A Selectric terminal provides a less expensive alternative, although the printing speed on such a terminal is about one-third of what can be achieved on the daisy-wheel printers.

Software: Where the Power Lies

Although the basic capabilities of the word-processing system depend to a large degree on whether the system is disk-based or uses cassette storage, the real power in the operator convenience and capability of the word-processing system lies in the software. Word-processing packages available today for microcomputers cover the field from simple string-handling text-editing packages to very sophisticated packages for professional use. The capability of the software defines how useful the system is in allowing the operator to perform a number of functions.

Editing: A word-processing package should provide for easy editing of particular lines of text, both in the initial creation of the text and on subsequent review and editing. One useful feature in editing is the ability to delete or insert characters and words from existing text without having to retype the entire line of text. A number of the BASIC interpreters used on microcomputers (such as Microsoft BASIC) have line editors that can be used by simple text-editing systems. The more sophisticated packages contain their own in-line text-editing capabilities which allow for extensive input editing.

Movement of Text: The ability to move blocks of text from one place in a document to another differentiates simple text-editing systems from real word-processing. In dealing with large documents, the user frequently finds the necessity of moving sentences, paragraphs, or larger blocks of text to other places within a document. A useful word-processing system should provide this capability in a simple fashion without the need of elaborate copying or other operations.

Access to Other Documents: The ability to access previously created

Electric Pencil

The Electric Pencil Word Processor is not included in this section, but will be reviewed in a future issue of *Creative.* Briefly, the Electric Pencil is a screen-oriented word processor with sophisticated features for both text manipulation and printing on a hard-copy device. It is available on Tarbell and CUTS cassette, and North Star and CP/M compatible diskette. From what we've seen so far, the Electric Pencil looks like an excellent word-processing system, suitable for both personal and business use (several OEMs are using it, too). It costs $100 for a standard printer version, $150 for a Diablo HyTerm version, and $15 for the manual. For more information, contact Michael Shrayer Software, 3901 Los Feliz Boulevard, Los Angeles, CA 90027.

"Technology is the means by which man extends his potency in manipulating his environment. Most tools are muscle multipliers — the shovel, the gear, the jet. The mind multipliers are language, the book, film, television, and now the computer. But the computer is qualitatively different — it is responsive. Children have a natural affinity for it."

Dean Brown
Stanford Research Institute, 1971

The printer...is probably the most critical item in determining whether the system produces truly useful output.

documents and copy selected portions of their text into a new document is an extremely useful feature when the word-processing system is being used in any type of production environment. The simplest approach is to provide for the copying of an entire document with the subsequent deletion of the unneeded portions. The more sophisticated approach is to be able to access, list, and copy from an existing document.

Search and Modify: The flexibility of word-processing systems is increased by their ability to scan entire documents looking for particular character strings. Frequently coupled with this feature is the ability to modify existing character strings where found in the document. For example, a user may wish to change the string "no. 3758" to "Stock number 3758-H" throughout a long document.

Formatting and Justification: A basic function of a word-processing system is to be able to produce documents with a large degree of control over the format of the completed document. More sophisticated word-processing systems can vary the margins and otherwise alter the format of a document regardless of the form in which the document is stored. Simpler text-editing systems will frequently force the user to define the format as the document is typed in or is later modified. Many text-editing systems will provide the ability to define a right margin and cause lines to be continued on the next without crossing the margin. Some systems will insert additional spaces or even proportionately space the text in order to justify to the right margin. In addition, the power of a system should be judged by the ability to do other types of formatting, such as indenting text from both the left and right margins and having flexible tabulation.

Other Fancy Formatting: The more powerful word-processing systems will provide the user with a number of useful features which may be superfluous to the occasional user. Included in this group are automatic pagination and page numbering, headings on each page automatically inserted, the ability to reserve "floating space" on a page for the insertion of diagrams or photographs, subscripting and superscripting for footnotes, and the inclusion of footnotes within documents. In addition, at least one word-processing system contains extensive controls for the user to insert text from a terminal or other documents as the primary document is being processed. In addition, this system provides the user with the ability to define a logical "if" test for changing control within a text document.

Expandability of Systems

A primary consideration for a number of users of word-processing systems on microcomputers should be the ability to expand the system to accomplish additional tasks. Consideration should be given to whether the word-processing system is amenable to using additional input terminals and high-speed printers for preparing drafts. In addition, a few word-processing systems will produce output directly usable by phototypesetting machines to produce cameras-ready copy for printing. Some users will find helpful data communication features available on a few systems.

Conclusion

The label "word processing" is applied to a wide variety of text-editing systems available on microcomputers. The user considering the purchase of a microcomputer or a word-processing software package should look closely to determine whether the particular package is really well suited for his application. The very-low-cost text-editing system using cassette storage may be quite adequate for the occasional user. The availability of CRT displays may greatly increase throughout. In addition, the user who plans to use word-processing in a production-type environment may find that the ability to expand the system is an important consideration. ■

business computing

WORD PROCESSING: Technical Systems Consultants

This world is producing millions of words of text each day ... and they all need to be organized before publication. It would certainly be a never-ending task if it all ... were to be done manually. It simply would not get done.

This world is producing millions of words of text each day. There are words in newspapers, magazines, books, catalogs, pamphlets, letters, documents, and manuals, and they all need to be organized before publication. It would certainly be a never-ending task if all text formatting and organization were to be done manually. It simply would not get done. Thanks to computers and programs called text processors, text formatting (sometimes called word processing) becomes a fairly trivial task. The text processor allows for convenient and precise page formatting and organization. The final copy becomes extremely readable and neat, which are desirable features of any printed matter.

Just what can be done with text processors? The simplest functions perform exact page-fitting. In other words, if the text page should have one-inch margins with a page number centered at the bottom of each page, and perhaps a special title at the top of each, the processor will automatically provide these, given the appropriate commands. Line justification is another feature provided. Several types are available which include left-hand justification (left edge straight, right edge ragged), right-hand only justification (left ragged, right straight), left and right (both edges are straight), and center justification (both edges ragged but lines centered). An extensive text processor will provide features that will allow special operations such as footnote processing. The TSC Text Processing System supports all of the above features.

To gain some insight into the use of a text processor, several specific examples will be given using the TSC Text Processor's command set. The commands used by text processors vary from system to system but many are used in the same fashion. The TSC Text Processor uses an intermixed command and text method. To issue a particular command to the processor, it is necessary to start the command in column number one of a new line and begin the command with the control character, a period (.). This is the method used by most of the large-scale system formatters including NROFF (a text-formatting program written at Bell Laboratories; it runs on many large operating systems, including the UNIX Time Sharing System), which is the system the TSC Text Processor has been modeled after.

Environment

Before any specific examples are shown, a description of the "environment" will be given. The environment refers to the basic page and formatting features which will be in effect unless otherwise specified. The initial or default environment is very important. The TSC processor, without any command information, will perform left and right justification with a line length of 65 characters (the standard 8½-inch page line width). Page length is initially defined to be 66 lines, which is the standard for 11-inch paper and 6 lines-per-inch spacing. Other initial environment features provide for the passing of blank lines to output, and for any line starting with a space or spaces to create a new line with the spaces now treated as unpaddable space characters, which are characters that appear as spaces on the output but are not recognized as such by the processor. (This means these spaces will not be "spread out" by the justification routines.) With the environment initialized as above, it is possible to take any text file not having special command information embedded in it and still receive a sensibly formatted output. This is an important feature often overlooked by many processor designers. The environment may, of course, be changed or modified at any time by the use of special commands to allow for more personalized and detailed formatting jobs.

Commands

Let's take a look at some specific commands of the TSC processor. One of the simplest commands is the center lines command, . CE N, where N is the number of lines to be centered within the current line width. To use this command, as with any of the commands, it is only necessary to place the command right before the lines it is to affect. For example:

```
. CE 2
The Design of Text Processors
An Introduction
```

will cause the two lines listed to be neatly centered on the page. It can be seen that this is much easier than trying to manually calculate the correct spacing.

The initial environment was previously described. All the

parameters may be easily changed by the commands which directly affect them. One of the commands is . LN N and sets the current line length. To set the line length to 50, all that is necessary is a command line which reads as follows.

. LN 50

The length is now 50 columns. Another parameter easily set is the page length using the command . PL N, where N is the number of lines per page desired. Some other commands which change environment parameters include . FI and . NF which turn fill mode on and off (no fill) respectively. Fill means that as many words that will fit within the current line length are placed on each output line. This gives a straight left text edge and a slightly ragged right one. *No fill* simply copies the input lines directly to the output. It should be noted that 'fill" must be on for any justification to occur. The justification feature may be turned off using . NJ for "no justification" or the type of justification may be set using . JU X. The X is the selection character and may be null, which turns justification on in the mode it was previously defined; it may be R for right hand, C for centered, or N for normal (left + right). Left justification is obtained by turning "fill" on and justification off.

Capitalization

Another environmental parameter is the capitalization mode. This is a special feature found only on the TSC Text Processing System and allows an upper-case-only terminal to be used for preparation of text that will later be output on a hardcopy device having lower-case capabilities. The commands . CP and . NC turn this feature on and off respectively. If this mode is on, all letters are automatically converted to lower case unless preceeded by a @. The @ should be thought of as a typewriter shift key in its function. Another feature also enabled in this mode is similar to the "shift and lock" on a typewriter. By typing a ↑ all characters following will be upper-case until another ↑ is encountered.

It is often desirable, for readability, to use multiple spacing between lines. The TSC processor will allow this using the command . MS N where N is the space count desired and defaults to double spacing (N=2) if no value for N is given. The single-space mode can be restored by either using . MS 1 or . SS for single space.

Margins and Indentation

Another group of commands deal with left margins and indentation. The left margin is normally set to 0 since the output device usually provides its own left margin (determined by paper positioning). Some applications require a wider margin, at which time . LM N may be used to redefine it to be N spaces wide. Indent is similar to the left margin control with one difference. . LM N preserves the line length and simply moves the line to the right N spaces. . IN N, on the other hand, effectively reduces the line length by N columns to preserve the right-hand margin. Setting the indent back to 0 will restore the full line length. Another form of indenting can be done by the use of the single indent command . SI N. Single indent is identical to indent except it is automatically restored to 0 after the line is output. It should be noted that the commands for left margin, indent, and single indent, are additive in that if the following string of commands is issued:

. LM 10
. IN 8
. SI 5

the resultant output line would be preceded by 23 spaces, succeeding lines are preceded by 18 spaces assuming another . SI command was not used.

Caution A note of caution is necessary concerning a characteristic of several of the processor's commands. Most commands will perform only their specified function but some also cause a line "break." A break is the forcing of output of the line currently being collected in the line buffer. Normally a line is kept in the buffer until the specified line length has been reached, at which time justification may or may not occur, depending on the mode enabled (also assuming that "fill" is turned on). The break will cause the partial line to be output without being filled, but the appropriate justification will be performed. This is useful for starting new paragraphs or new blocks of text. Some of the commands that cause a break are . CE, . FI, . NF, . IN, and . SI. Sometimes it is desirable that these commands do not cause a break. This can be done by using the "no break" control character, :. So far, all commands have been shown preceded by the normal control character, a period. To set an indent of 10 and not cause a break, the following should be used:

: IN 10

The colon may be used with any command, whether the command normally causes a break or not.

Blank Lines

It is often necessary to produce a section of one or more blank lines. The space command, . SP N, can be used to output N blank lines. The space command also causes a break. If N is not specified, the processor will output one blank line. It may be required that the blank lines all be on the same page, maybe for later insertion of a photograph or illustration. The TSC Text Processor allows this by using the

"save space" command, . SV N, where N is the number of lines required. If there are not N lines remaining on the current page, no line is output but instead, printing continues and the count (N) is saved for later use. When the next page is reached, the "output saved space" command may be used, . OS, which will then produce the remembered number of blank lines. A convenient method for using . OS will be given later. Another similar command is the "need lines" command, . NL N, where N is a line count. This command says that there must be N lines remaining on the current page, and if there are not, eject to the next page. This is convenient for keeping special blocks of text together (keep them from being split by page boundaries), or for not starting a new paragraph at the bottom of a page if only one or two lines will fit.

Page Titles

The commands described so far will allow very nice page formatting. If these were all that were available in a text processor, much time and effort would be saved. The TSC Text Processing System, however, offers many more commands and much more versatility. One feature often needed in documents or manuals is the page title. There are many different ways of providing titles but the TSC processor uses a title command which has the form:

. TL 'field1'field2'field3'

where field1 is left-adjusted, field2 is centered, and field3 is right-adjusted. Any one or all of the fields may be present. Another feature supported in the TSC processor is the ability to print the current page number in the text. Any place a percent sign (%) appears, it will be replaced by the page number. A few examples will clarify the use of the title command.

```
. TL 'Main Title'''
. TL ''Centered Title'Date'
. TL ''-%-''
```

The first line will left-adjust "Main Title" on the page. The second example causes "Centered Title" to be centered and "Date" to right-adjusted. The final example will cause the current page number to be printed between two dashes.

Now it would be nice if there was some way of getting the title (and maybe a few other commands) to automatically execute at the top and/or bottom of each page of output. The TSC processor offers two advanced features to perform this task: macros and traps. A macro is a set of commands grouped together and given a name. When this name is later referenced, the entire group of commands will be executed. Essentially, what results is the ability to write special programs using the command set of the processor to do specific tasks such as headers, paragraphs, special titles, etc. The trap allows the user to specify a certain line on the output page where a specific macro is to be executed. To solve the title problem stated above, it is convenient to define two macros, a header macro and a footer macro. The purpose of the header is to perform a sequence of commands to make the top of each new page appear the same. The footer macro works at the bottom of each page. Suppose it was required that the top of each page have three blank lines followed by a centered title and the bottom of each is to contain a centered page number between dashes. The following macros and trap placement would satisfy this requirement.

```
. DM HD
: SP 3
. TL ''Page Title''
: SP 3
. .
. DM FT
: SP 3
. TL ''-%-''
: PG
. .
. AT 1 HD
. AT -7 FT
```

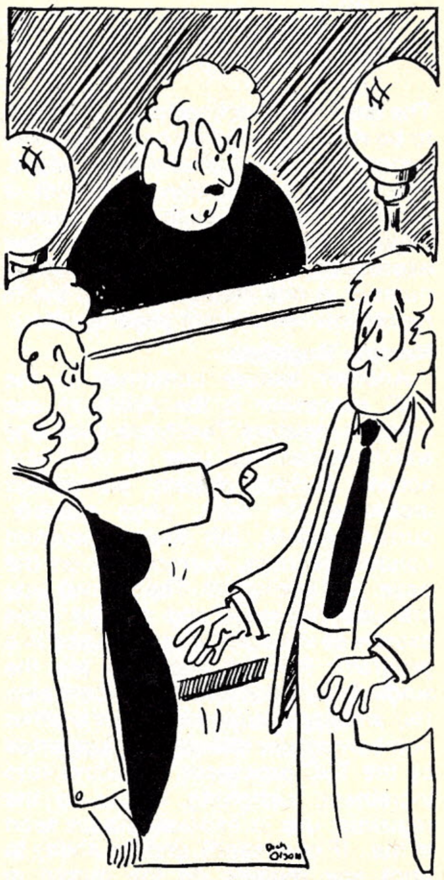
The . DM command defines a macro and the first one listed in the example defines the header macro called HD. The header macro will space down 3 lines (without causing a break since the no-break control character(:) was used), print a centered title, and finally print 3 more blank lines without causing a break. The last line of the header macro definition is ". ." and is the command for terminating a macro definition. The second macro defined is FT and is used for the footer. Upon execution it will space down 3 lines (without a break), print a centered page number, and eject to the next page. The . AT commands were used to set the trap locations. . AT 1 HD causes the header macro to be executed at line 1 of every new page while . AT -7 FT causes the footer macro FT to be executed at the 7th line from the bottom of each page. The ability to specify trap locations and define macros makes titles and footers extremely simple and efficient.

Changes

One of the important aspects of using a text processor is the ability to make a few minor command changes and greatly change the final copy. As an example, suppose at the last minute it was decided that it would look better if there were four blank lines at the top of each page rather than three. If the document were being prepared by hand it would be necessary to retype the entire work to obtain the extra space. Using a small text processor it would only be necessary to go back and change the line count before each title. The TSC Text Processor and its ability to define macros means only one line in the entire text file needs to be changed. The second line of the header macro, which is currently ": SP 3" would be changed to read ": SP 4". One simple change and the desired result is obtained! It should be kept in mind that when preparing documents with a processor supporting macro capability, all of the often-used command strings should be defined in a macro so simple global changes may be easily performed if so desired.

Thanks to computers and programs called text processors, text formatting (sometimes called word processing) becomes a fairly trivial task.

D.P. DOODLES



"Relative Pointer"

Conditional Commands

There are more advanced features supported in the TSC Text Processing System. One of these is the ability to do conditional command execution. There are four forms of this command:

```
. IF O . XX
. IF E . XX
. IF N. XX
. IF !N . XX
```

where O and E stand for Odd and Even page number respectively, and N can be a number, a number register (to be explained shortly), or an expression containing numbers and number registers. The exclamation point is the NOT operator and . XX is any command or macro name. The command works as follows; IF the condition is true (page is odd or even, or the number or expression is greater than zero) the command XX is executed, otherwise it is not. Preceding the expression by ! will cause the command or macro to be executed only if the condition is not true (less than or equal to zero). The following special header macro definitions will illustrate the use of this command.

```
. DM HD
: SP 3
. IF O . TL '''Title'
. IF E . TL 'Title'''
: SP 2
. .
. DM HD
: SP 3
. IF %-1 . TL "Title"
: SP 2
. .
```

The first header defined causes the title to be right-adjusted on odd-numbered pages and left-adjusted on even pages. The second definition will print a centered title on each page except page number one since the value of the expression will be zero when the page number is one (remember that the % represents the current page number).

Number Registers

Another feature contained in the TSC processor is the ability to use number registers. Two types exist, one which allows the user to read and access certain system parameters including the date, page number, current indent, left margin, current column position, current line on the page, and line length. The second type are user-definable and can be used exactly as variables would be used in a program. Number registers are the single letters A-Z and the percent sign (%) already introduced. Several other number register features are supported by the TSC processor, including auto increment, assigning values to the registers, use in expressions (as seen in the . IF command), and the ability to print any register on the output in either Arabic, capital Roman, or small Roman numerals.

Some processors, including TSC's, allow communication between the processor and the operator during actual text processing. Three of these commands take on the following form:

```
. ST
. TM any string
. GI any string
```

The first command will stop the processing and print STOP on the user's terminal. This may be desirable if special paper positioning is required or other special action is needed. When the processor has been stopped it may be restarted by typing any character on the terminal except an S, which will halt processing. The second command listed will send "any string" to the terminal as a special message. It may be used before the STOP command to issue special instructions to the operator. The last command will "Get Input" from the terminal and insert it into the output stream. "Any string" can be used for a prompt. An example where this command is quite useful is in the preparation of form letters. The processor may prompt the operator for names and addresses which are then typed in at the terminal and automatically inserted into the text!

Divert Text

One final command will be described in this introduction, the "divert text" command. Sometimes it is desirable to save text currently encountered for later use. An example of this is when trying to do footnotes. It would be nice if immediately after the footnote reference was made, the actual footnote text could be typed, but saved for later insertion at the bottom of the page. The mechanism which allows this sort of operation is called a "diversion" and is only available on the more complex text processors, such as TSC's. Two forms of the diversion usually exist:

```
. DI XX
. DA XX
```

where . DI instructs the processor to divert the following text into a diversion space named XX and . DA says to divert and append to the diversion space named XX. During diversion, all normal text processing still takes place, but rather than outputting the text to the printer, the text is written to a special place internal to the processor. The diversion process continues until the command for a divert is found without a name specifier. To recall the diverted text, it is only necessary to call it by name, exactly as macro calls are performed.

Macro Set

As an advanced exercise and demonstration of the diversion process (as well as many other processor commands) a complete set of macros for handling footnotes will be described. The reader should note that the following example is very complex and several readings will probably be required to fully understand its operation.

```
. NR B 7
. DM HD
: SP 2
. IF %-1 . TL 'FOOTNOTE TEST'''
: SP 2
. AU 1
. NR X 0
. NR W 0-#B
. IF #V . TR
. NS
. .
. DM FO
. NR V 0
. IF #X . FT
. CH FO -#B
: PG
. .
. DM NM
. TL "-%-"
. .
. DM BF
. DA TX
. EV 1
. IF !#+X-1 . SA
. DM EF
. BR
. EV 0
. DI
NR W -#V
. CH FO #W
. IF #N-#P-#W . CH FO #N+1
. .
. DM SA
--------------------
. BR
. .
. DM TR
. BF
. NF
. FE
. FI
. EF
. .
. DM FN
. DI FE
.
. DM FT
. EV 1
. NF
. TX
. RM TX
. DI
. FI
. EV 0
. .
. AT 1 HD
. AT -#B FO
. AT -4 NM
. CH FO 70
. AT -#B FN
. CH FO -#B
. EV 1
. AU 1
. LN 55
. EV 0
```

This example is quite similar to the one given in the "NROFF Users' Manual" written by J. Ossanna, of Bell Laboratories. To use these macros, merely insert their definitions at the beginning of the text file, and immediately after a footnote reference has been made, call macro BF. Following the call, simply type the footnote text and end it with a call to EF.

A description of the macros follows. The first line defines number register B and sets it equal to 7. Number register B is used to specify the size (in lines) of the bottom margin. A header macro definition follows (HD) and provides several functions. After spacing down two lines, the title is output unless it is page number one (the IF command). Two more lines are produced and the auto-increment value is set to one. Number register X is cleared and it is later used to keep track of the number of footnotes on the current page. Next, W is set to the location of the bottom margin trap and will later be adjusted as necessary if footnotes are added. The IF #V command checks to see if there is any remaining footnote text from the previous page and if so they are reprocessed (number register V contains the line count of the last diversion). Finally, the no-space mode is turned on the suppress any spaces that might otherwise get printed needlessly at the top of the page.

The footer macro, FO, clears the diversion count, V, and checks the value of X. If X is not zero (meaning there were footnotes on the page), macro FT is invoked. The footer is then restored to its original location by using the Change command as defined by B. The last command does a page eject. Macro NM prints a centered page number at the bottom of each page.

The begin-footnote macro, BF, starts with a divert append into the diversion space called TX. The environment* is switched, and if it is the first footnote on the page, macro SA is invoked, which outputs a set of dashes as a simple footnote separator line. Diversion of the footnote text continues until macro EF is called. At this time a "break" is executed and the original environment is restored. The diversion is stopped with the DI command. Number register W is updated by the number of diverted lines and the footer trap line is changed to compensate for the added footnote lines. Finally, if the number of diverted lines was great enough to move the footer trap up past the current line position, the trap is reset to the next line. TR is responsible for rediverting any lines of footnote text which will not fit on the page. It is very

*Environment switching is a feature supported by many of the larger text processors (including TSC's), which allows all of the major environment parameters to change simultaneously.

An Example

As an example of how a text processor can be used, a sample section of text is shown first with the commands and then as the text processor would output the final copy.

```
. SP 2
. CE 2
TEST OF SEVERAL
PROCESSOR COMMANDS
. SP
. SI 5
@THIS EXAMPLE SHOWS HOW COMMANDS AND TEXT CAN BE INTER-
MIXED FOR LATER PROCESSING BY A TEXT PROCESSOR.
@THE EXAMPLE STARTED BY CENTERING TWO LINES FOLLOWED BY A
SINGLE INDENT TO SIGNIFY THE START OF A PARAGRAPH.
@THE CAPITALIZATION MODE IS ON AND THE UPPER CASE SHIFT
CHARACTERS ARE BEING USED.
. SP
. LM 10
. LN 45
. JU C
@THE ADJUST MODE WAS JUST CHANGED TO CENTERING AS WELL AS
A LINE LENGTH OF 45.
@THE LEFT MARGIN WAS SET TO 10 TO GIVE A NICELY CENTERED
NARROW LINE.
@SPECIAL EFFECTS LIKE THESE ARE EASILY ACCOMPLISHED.
. SP
. LM 0
. LN 65
. JU N
@THE PARAMETERS WERE JUST SWITCHED BACK SO THE LINE
APPEARANCE WILL BE RESTORED.
THIS IS A SHORT EXAMPLE BUT SHOULD SHOW HOW THE COMMANDS
CAN BE INTEGRATED WITH THE TEXT.
```

This example appears thusly in its expanded form:

```
                    TEST OF SEVERAL
                  PROCESSOR COMMANDS
This example shows how commands and text can be intermixed for later
processing by a text processor. The example started by centering two lines
followed by a single indent to signify the start of a paragraph. The capitalization
mode is on and the upper case shift characters are being used.
             The adjust mode was just changed to centering
                as well as a line length of 45. The left
                   margin was set to 10 to give a nicely
                centered narrow line. Special effects like
                     these are easily accomplished.
The parameters were just switched back so the line appearance will be restored.
This is a short example but should show how the commands can be integrated
with the text.
```

unusual for this to happen but this may occur if a footnote is very long and is referenced near the bottom of the page.

Macro FT is used for reading back the diverted text. It switches environments, sets the no fill mode, and calls TX, the actual footnote text. TX is then removed from the macro list, the fill mode is restored, and the environment switched. The last group of lines is used to define the trap locations of the various macros. The header is set to line one, and NM is set to execute four lines from the bottom of the page. The trap for the footer is planted at -#B, then moved past the bottom of the page while FN is also placed at -#B. FO is then moved back as originally placed so in effect both FO and FN are placed at the same line, but trap FN can only occur if the footer trap is moved up by the occurrence of a footnote. The last lines switch to environment one and initialize it for a line length of 55 and auto increment of one.

This introduction to text processing is intended to be only that and is not a complete treatment of the subject. Many commands and features have been omitted. The ones included are the most general and the most used commands which offer the user a great deal of control and flexibility. Hopefully some eyes have been opened to the wide variety of applications of the text processor.

Availability

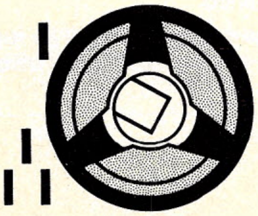
The TSC text-processing system manual, containing an introduction, summary of commands, reference manual, macro descriptions, and a commentated listing of the assembly-language source program, is $32.00. A cassette tape of the object code, available in several formats, is $6.95; the same in paper tape is $8.00.

Text Editing System

A companion to the TSC text-processing system is their text-editing system, for text manipulation at the level of word, line and content.

Editing commands permit additions, changes and deletions where desired, moving blocks of lines, setting tabs, etc., and for writing part or all of a text onto cassette or paper tape.

The text-editing system, with source program, is $23.40; cassette is $6.95, paper tape $8.00. ■



business computing

WORD PROCESSING:

INFO 2000 Corp.

Mike Busch and Ron Raikes

TEXT 2000 is a proprietary word-processing software package developed by INFO 2000 Corporation expressly for use with the INFO 2000 Business System. The package was written in Z80 assembly language and runs under the Digital Research CP/M disk operating system. The TEXT 2000 software is included in the basic price of the INFO 2000 Business System, and is not sold separately.

Overall Scope

The TEXT 2000 software is designed to address five principal areas of word processing in which most businesses and educational institutions need automated assistance:

1. *Correspondence:* TEXT 2000 can turn out immaculate film-ribbon originals of correspondence at the rate of more than 500 words per minute without ever making a typographical error.
2. *Editorial:* TEXT 2000 facilitates fast revision and retyping of updated versions of documents as fast as an editor can mark up the changes.
3. *Cut-and-paste:* TEXT 2000 can quickly assemble a collection of standard paragraphs into a final-typed letter, specification, or other document.
4. *Direct mail:* TEXT 2000 can maintain extensive lists of names and addresses, select subsets of these lists according to various selection criteria, and print mailing labels or address envelopes. TEXT 2000 can automatically type multiple originals of a standard letter, personalized for dozens or hundreds of addressees from a list.
5. *Typesetting:* TEXT 2000 can produce camera-ready copy with multiple text columns, perfect justified margins, boldface print, underlining, true proportional spacing, variable horizontal and vertical spacing, multiple type fonts, and other professional-looking effects which are not possible with manual typing.

Major Modules

The TEXT 2000 package is composed of three major software modules, all interfaced with a single function menu so that the user sees TEXT 2000 as a single integrated application. The three major modules are: (1) the text editor, (2) the text formatter, and (3) the mailing-list processor.

Text Editor

Although the TEXT 2000 text editor includes a powerful repertoire of editing capabilities, it is designed to be so simple to use and so easy to learn that a secretary can master its basic facilities after only an hour or two of practice. More sophisticated features can be learned gradually in stages.

The text editor is of the full-screen type. Instead of a complicated vocabulary of editing commands, the TEXT 2000 editor uses the twelve-key numeric keypad on the video console as a function key cluster to specify common editing functions such as cursor positioning and inserting, deleting or replacing a character, word or line of text. Less common functions are specified by typing simple English directives. The TEXT 2000 editor uses a small area in the lower righthand corner of the video display screen to continuously remind the operator what function has been specified (inserting, replacing, etc.) to reduce the chance of error if the operator is distracted in the midst of an editing operation.

Text is typed in as a continuous

Mike Busch and Ron Raikes, INFO 2000 Inc., 20630 South Leapeeod Ave., Carson, CA 90746.

stream of words without any concern about where one text line ends and another line begins. TEXT 2000 displays everything on a large video display exactly as it is entered. When typing reaches the righthand edge of the display screen, the editor automatically shifts the partially completed word to the beginning of the next line on the display.

The TEXT 2000 editor permits rapid browsing, detailed review, and flexible editing of text. Whenever characters or words are inserted or deleted, subsequent text on the display is pushed down or pulled up automatically as appropriate. Entire blocks of text may be moved freely from one portion of the document to another. All occurrences of any word or phrase may be quickly located, and any or all of these occurrences may be replaced with a substitute phrase.

Text Formatter

TEXT 2000 formats the text automatically as it is typed onto paper. Words are organized into lines of proper length in accordance with the requested form size and margin settings. Character spacing is automatically adjusted to provide flush justification of both left and right margins if desired.

True proportional spacing is provided by TEXT 2000. Proportional spacing recognizes that certain characters (such as i, 1) require less horizontal space on a printed line than other characters, and that some characters (such as m, w), need even more space. When used with a proportional-spacing typewheel, TEXT 2000 can produce extremely professional-looking camera-ready copy which has a typeset (rather than typewritten) appearance. TEXT 2000 permits proportional-spacing tables for a variety of type fonts to be stored on disk.

Automatic centering, underlining, boldface, pagination, page numbering and titling, tabulation, indentation, and numerous other text formatting functions are performed by TEXT 2000 in accordance with simple formatting directives which may be interspersed into the text. See the table for a summary of the available formatting directives.

Formatting directives are composed of simple English words which may be easily learned by any user with basic clerical skills. (For experts, all directives may be abbreviated to two letters.) Directives are surrounded by brackets to distinguish them from text, and multiple directives may appear in one pair of brackets if separated by slant characters. Unlike most other word-processing packages, TEXT 2000 permits form dimensions, margin settings, character spacing and line spacing to be expressed in terms which are natural to the operator: inches (to two decimal places), characters-per-inch, lines-per-inch.

Up to three lines of heading information may be specified, and will be automatically printed at the top of each page. Any heading line may contain a page number, which is calculated by the TEXT 2000 formatter. Five alternative heading line formats are provided: justified, centered, flush left, flush right, and alternating left and right. The latter format is perfect for copy prepared for two-sided reproduction.

Hyphenation is not needed in most applications, due to the capability of TEXT 2000 to justify margins by varying character spacing. However, hyphenation may sometimes be used advantageously to improve the readability of text which contains long words and which is to be formatted into narrow columns. The TEXT 2000 formatter provides for this as follows. The character tilde (~) may be used to indicate optional hyphenation points in a long word (such as micro~pro~cessor). The character hyphen (-) indicates a mandatory hyphenation point in a compound word (such as micro-pro-ces-sor-orien-ted). The double hyphen (--) is printed as a single hyphen, and may be used to prevent TEXT 2000 from breaking the hyphenated word across two lines (such as Phone 1--800--555--1212).

The TEXT 2000 formatter goes to considerable lengths to take maximum advantage of the exceptional capabilities of the word processing

---

TEXT 2000 is a proprietary word processing software package developed by INFO 2000 Corporation expressly for use with the INFO 2000 Business System. The package was written in Z80 assembly language and runs under the Digital Research CP/M disk operating system. The TEXT 2000 software is included in the basic price of the INFO 2000 Business System, and is not sold separately.

OVERALL SCOPE

The TEXT 2000 software is designed to address five principal areas of word processing in which most businesses and educational institutions need automated assistance:

1. Correspondence: TEXT 2000 can turn out immaculate film-ribbon originals of correspondence at the remarkable rate of more than 500 words per minute without ever making a typographical error.

2. Editorial: TEXT 2000 facilitates fast revision and retyping of updated versions of documents as fast as an editor can mark up the changes.

3. Cut-and-paste: TEXT 2000 can quickly assemble a collection of standard paragraphs into a final-typed letter, specification, or other document.

An example of how TEXT 2000 can set copy to any desired width: the opening of this article.

TEXT 2000 runs on the INFO 2000 Business System, as shown here.

TEXT 2000 Formatting Directives

Justify	Line Spacing = 1,2,...
Flush Left	Tabs = column , column ,...
Flush Right	Blank Lines = number-of-lines
Centered	Skip = number-of-lines
Variable Pitch	Page = page number
Fixed Pitch	Conditional Page = number-of-lines
Proportional Spacing	Indent = columns
Uniform Spacing	Undent = columns
Bidirectional Printing	Heading Line = 1, 2, or 3
Unidirectional Printing	Heading Format = format
Strike = 1 or 2	Heading Gap = number-of-lines
Continuous Forms	Switch
Single Forms	Text File = file-name
Horizontal Margins = left , right	Insert File = file-name
Vertical Margins = top , bottom	Get Insert
Form Size = width , length	Comment Line
Horizontal Spacing = characters-per-inch	Prompt Line
Vertical Spacing = lines-per-inch	

NOTE: All TEXT 2000 directives may be abbreviated to two letters. A one-word directive is abbreviated to its first two letters (for example, [in=5] is equivalent to [indent=5]). A two-word directive is abbreviated to the initial letters of each word (for example, [fs=8.5,11] is equivalent to [form size=8.5,11]).

printer (Xerox Diablo HyType II) which is offered with the INFO 2000 Business System. These capabilities include: bidirectional printing (one line left-to-right, next line right-to-left), proportional spacing, boldface printing, underlined printing, and continuously variable horizontal and vertical spacing of characters and lines. TEXT 2000 even provides a double-strike printing mode in which each character is struck twice; this provides film-ribbon quality using low-cost cloth ribbons, at a modest sacrifice in printing speed.

Mailing-List Processor

The TEXT 2000 mailing-list processor provides the capability to build and maintain files of names, addresses, and telephone numbers on disk. These files may be sorted (alphabetically, by ZIP code, etc.) as desired. They may be printed in the form of listings or pressure-sensitive labels, or may be used to address envelopes directly. And they may be used in conjunction with the TEXT 2000 formatter to produce multiple original typed copies of a standard letter, each addressed to a different addressee from the file.

TEXT 2000 mailing-list files contain, in addition to the usual information (name, address, ZIP, telephone), a special selection field which can be of a length and format chosen by the user. The TEXT 2000 mailing list processor allows sublists to be created by selecting only entries for which the selection field falls within a specified range or matches a specified pattern. This provides a flexible means for selecting names for a particular mailing.

Hardware Configuration

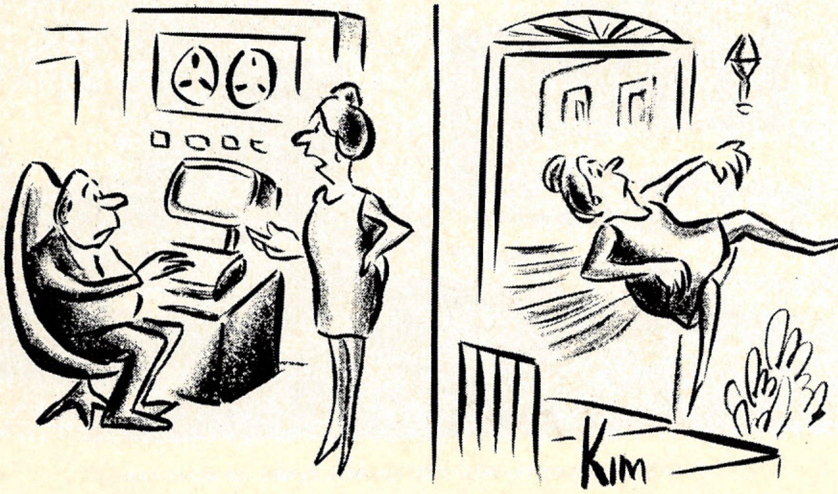
The TEXT 2000 software operates on the INFO 2000 Business System, a low-cost data processing system for small businesses, government entities, and educational institutions. A complete word-processing configuration including all necessary hardware and software can be purchased for less than eleven thousand dollars or leased for under $275 per month on a five-year lease.

This includes a Z80-based microcomputer with 36K of main memory (33K of RAM plus 3K of ROM), dual floppy disk drives featuring full-size 8-inch IBM-compatible diskettes and fast performance, a video console with a capacitive keyboard (no switch contacts) and a large pedestal-mounted display, and the Xerox (Diablo) word processing printer rated at more than 500 words per minute.

The basic price of the INFO 2000 Business System includes the TEXT 2000 word processing package described in this article. It also includes the CPA 2000 accounting package which provides Accounts Payable, Accounts Receivable, General Ledger and Financial Statement functions.

Summary

When provided with sophisticated applications software and suitable peripheral equipment, a low-cost microcomputer system can provide the speed and functional capabilities of dedicated word-processing systems (from IBM, Xerox, Lanier, and others) for thousands of dollars less cost. At the same time, such a microcomputer system does not have to be dedicated to word processing, but can also be used to handle accounting, computation, and other data processing needs. ■



"Either this goes or I go!"

business computing

WORD PROCESSING:

Ohio Scientific

Ohio Scientific's new WP-1 Word Processor is a full text editor which operates at both the character and line levels. The editor is complete with disk files and has formatted printout, which makes it ideal for writing letters, manuals, reports, and all normal, everyday business forms. The new Word Processor's disk files are completely compatible with the disk files used for source code by the Assembler/Editor, so the Word Processor's powerful editing commands can be applied to Assembler source code. The Assembler can actually be linked to the Word Processor as an option. The Word Processor is specifically designed for secretarial or other nontechnical use. The Word Processor disk boots in automatically when D is typed, and asks "MEMORY SIZE?" The user simply types <return> (the Word Processor will then use all the memory available in the computer). The Word Processor features a very elaborate error-checking routine, which makes it virtually crash-proof. It instructs the operator as to what he did wrong via short messages. For you to gain an appreciation for the power of the Word Processor, let us discuss its five classes of commands.

Line-Editing Operations

The Word Processor has several line number-oriented operations. Each line is assigned a number between 1 and 65,000. It is not necessary for a user to specify a line number when entering text or modifying text. He can avoid the use of line numbers by simply typing 0 for any line he enters. The Word Processor will then automatically assume the line numbers are sequential after the last numbered line printed out. The user can print out any line or combination of lines by use of the P (print) command. By typing P<return> he will list the entire file in his workspace. Or he can specify certain lines by any of the following formulas:

mmm-nnn	lines from mmm to nn inclusive
mmm-	lines from mmm to end of file
-nnn	lines from beginning of file to nnn
mmm	line mmm only

This allows the user to examine any section of the file. It even has an R<return>or RESEQ<return> which causes a resequence of line numbers starting with line 10. An offset after the RESEQ command will cause the file to be resequenced with an offset, such that R1000 starts the resequencing with line 1000. After the resequencing is completed, the Editor will report the next available line number.

The line-editing portion of the Editor has two additional extremely powerful commands, T (transfer) command and the M (move) command. The T command will transfer any block of lines to a new position specified after any given line, without eliminating the original block of lines. For example, a file may have twenty lines specified as lines 10-200. By use of the command T100-200, 5 lines 100-200 will appear before line 10 as well as where they did originally, so that the file will now be thirty lines long. Line numbers are stripped off in the transferred file, so that these new lines each acquire a line number of 0. Via the RESEQ command the file can be resequenced such that new line numbers are reassigned to all lines. This command is useful when blocks, or portions of a file will be very similar. The user can transfer similar portions into different areas of the file, then use the powerful character editing operations to make any slight changes necessary for the new portion.

The Word Processor is specifically designed for secretarial or other non-technical use.

Character-Editing Operations

The Word Processor also has a powerful class of character-oriented commands. Two of these commands are the C (change) and F (find) commands. The C command is the same as the Edit/Replace commands in some timeshare systems. It allows you to change any string of characters, either on a single line, a block of lines, or in the entire working file. For instance the command C"e", "ee"<return> will change the character e in the file to the character ee wherever it occurs. This command can be employed in form letters where only a person's or company's name may vary from letter to letter, for example. The C command can have an optional line specifier, so that if a single line or block of lines is specified, the command will operate only upon that line or block of lines. The C command is thus capable of both local and global operations.

To complement the C command, the Editor also has an F command, which will find any string specified within

quotes and print out all lines containing that string within the specified range. With the F command we can search for all occurrences of a string within a block of lines, or within the entire file. This would be useful when looking for a particular name or word which may have different meanings in different contexts. We would thus be able to change the word in only one of its contexts, if desired.

Disk Commands

As stated above, the Word Processor is designed for use by inexperienced operators. Therefore it has its own internal GET and PUT file commands, which do not require the user to leave the Word Processor. The syntax for both the GET and PUT commands includes the naming of disk drive A or B and the starting track of the file. The Word Processor comes on a standard OS-65D diskette, where the first nine tracks are utilized by the Word Processor and the system. This leaves sixty-eight tracks available for files for the Word Processor, which should be placed in drive A of single or dual drive systems. Seventy-six tracks are available on drive B for use as Word Processor files. An individual Word Processor file is limited to the length or size of available memory.

Formatted Output Commands

The Word Processor has a powerful formatted output mode which is similar to runoff programs found in many larger computer systems. The formatted output is based on the L (list) command. This command lists text in the workspace without line numbers in justified form. It is used to produce Ohio Scientific's Small Systems Journal and the Word Processor manual. It performs both left and right justification to a character width specified by the user. After typing L, the user can designate a justified text width from twenty to seventy characters. He can optionally add a line specifier so that he can limit the output of justified text to a certain block of lines. If he does not specify any lines, then the entire file is listed in justified format. The L function

The practicality of the Word Processor is made possible by its use in conjunction with the Disk Operating System. Any program or text you compose may be stored on a diskette and saved for future use.

searches through text for the wordspace just under the total number of characters specified. It adds additional spaces between words on an equal basis to expand the line of text to the designated width.

In addition to the justified output and line specifications, the user can specify page format by also typing an F (format), followed by a page number, in the line with the line command. The presence of a format subcommand in the line command causes the output to page; that is, place spaces between pages of text. It adds a page number in the bottom of each page, starting at the page number specified. If the command is F10, then the starting page number is 10, and so on, up to 65,000 pages.

File Handling under WP-1

The practicality of the Word Processor is made possible by its use in conjunction with the Disk Operating System. Any program or text you compose may be stored on a diskette and saved for future use.

Each diskette contains 77 tracks, most of which are available for storage purposes. These tracks are arranged in concentric circles resembling the bands of a phonograph record. In a single drive system, the first nine tracks on the diskette are reserved for the Operating System. These tracks are intended to remain permanently on their tracks because they are essential for the operation of the system as a whole. The remainder of the diskette is clear and available for file storage. On a dual drive system, the upper disk (Disk A) has the necessary modules for the system. The remainder of Disk A and all of Disk B are available for storage. ■

Availability

The WP-1 Word Processor System is available on 8-inch floppy diskette, at $79. The WP-1A diskette, which includes both the Word Processor system and 6502 Assembler, is $99. Ohio Scientific Inc. is located at 1333 S. Chillicothe Rd., Aurora, Ohio 44202, (216) 562-3101.

business computing

WORD PROCESSING: Peachtree/Altair/MITS

John B. Hayes

Text-editing systems that operate on microcomputers cover a broad range from simple cassette-based systems to elaborate and expensive word-processing implementations. The Peachtree Word Processing System is a sophisticated system with features designed for the heavy user of word processing. The package is available through computer retailers throughout the United States under the terms of a limited-use license for the payment of a one-time license fee. In addition, the package is sold through Pertec Computer Corporation under the names Altair Word Processing Package and MITS Word Processing. The package is owned and distributed by TCS Corporation which operates retail computer stores under the name The Computer Systemcenter.

Line Numbers

The Peachtree Word Processing System operates on microcomputers that use 8080, 8085, and Z-80 microprocessors. The system requires the use of a CRT video display and at least dual floppy disks. The system is also available for Altair hard-disk units. The Computer Systemcenter recommends the use of either Qume or Diablo daisy-wheel printers to produce quality output.

The Peachtree Word Processing System is based on the concept of each line of text in the document being assigned a number in the range .001 to 999.999. Decimal line numbers allow the user to insert large amounts of text without the necessity of renumbering subsequent lines in the document, a procedure which preserves existing line numbers for reference back to previous drafts of the document. Each document in the system may contain up to 1,000 "lines" with each line holding up to 120 characters; the manner in which a line is stored in the permanent storage is not directly related to the line width used for printing. Therefore, a document in the system will hold approximately 40 pages of single-spaced text. The documents can be linked together so that automatic page-numbering continues from one document to another. In this manner, there is no limit to the size of a completed text document when a number of smaller documents are linked together.

Standard Formats

The basic formatting of a document can be specified within each document or simply by using a set of standard system formats defined by the user. In this manner, standard documents on a system can be defined as 60 characters per line, 55 lines per page, and single-spaced. A particular document can contain a command that specifies double-spacing from that point until a single-spacing command is encountered. All of the formatting commands in the document are visible to the operator and can be manipulated in exactly the same fashion as text. At the time that the document is printed, the formatting commands are operated upon by the system to have the desired result produced.

Accessing Other Documents

The Peachtree Word Processing System can access other word-processing documents through the use of a STAR file. The STAR file is simply a text document that has previously been created and is called upon by the user. All or any portion of the STAR file can be listed or copied to a new file. In addition, the STAR file can be searched for the occurrence of certain characters or words. However, the STAR file is protected from modification and cannot be overwritten, so a master document can be maintained as a STAR file and its integrity preserved. Any number of STAR files may be used, one at a time, in editing or printing documents. STAR files are frequently used to contain boiler-plate contract language, name and address lists, or other information which may be referenced during the creation or printing of documents.

By using the logical operator, and the field and record features of the STAR file, such applications as mailing lists can be built within the word-processing system.

Editor

Documents are created and edited in the system for the use of the EDITOR, which allows documents to be typed in and later retrieved and edited. The EDITOR provides extensive capabilities for moving or copying text within a document or from a STAR file to a document. The EDITOR allows a document to be searched for character

John B. Hayes, Alston, Miller & Gaines, 1200 Citizens & Southern National Bank Building, Atlanta, GA 30303.

strings, replacements made where found, lines to be inserted or deleted where needed, and lines to be renumbered or listed. In addition, the EDITOR provides for very convenient input editing on the CRT by allowing existing lines to be expanded for the insertion of text, and characters, words, or portions of lines to be deleted.

Additional Features

The Peachtree Word Processing System contains a number of additional features for use in heavy-volume word-processing applications. For example, the QUERY command prints a message on the screen and waits for the operator to provide the necessary text (or other commands) which are to be input in the document at that point. The system also has a number of variables which can be defined at the time the document is run for "fill in the blanks" operation. The variable DATE inserts the current date in a document. These features are useful for repetitive letters and other work from standard documents. The system also provides the logical IF test which allows two text strings to be compared to see if the second string contains, excludes, equals, or does not equal the first and allows for a change in control to be executed at that point. In this way, documents may be "programmed" to check for the occurrence of certain information. By using the logical operator, and the field and record features of the STAR file, such applications as mailing lists can be built within the word-processing system.

Existing documents can be taken from the word-processing system and used to drive other equipment, such as phototypesetting machines.

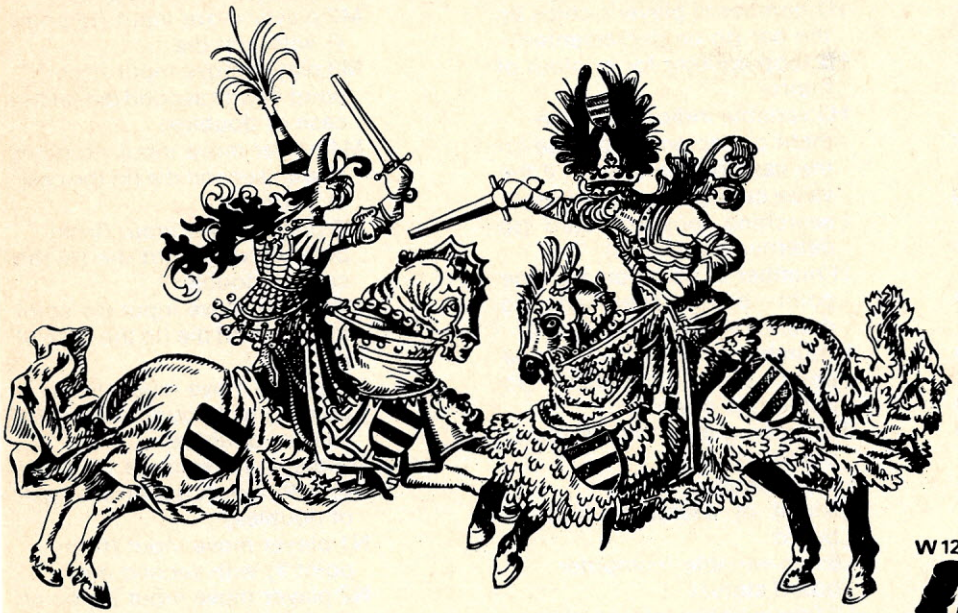
The Peachtree Word Processing System contains all of the usual editing features such as the ability to center lines, underlining, overprinting of characters to produce bold print, filling from one text line to another, providing for floating spaces to allow the insertion of diagrams in text, the ability to define a heading which is to be located on each page, control over the placement of page numbers, right justification, indenting both right and left, superscripting and subscripting, and complete tabulation, including decimal tabulation for accounting or statistical work.

The file structure of the word-processing system is designed so that documents in the system can be accessed through user programs written in BASIC. This feature allows such additional capabilities as the user generating text information from other programs, such as accounting programs. In addition, existing documents can be taken from the word-processing system and used to drive other equipment, such as phototypesetting machines.

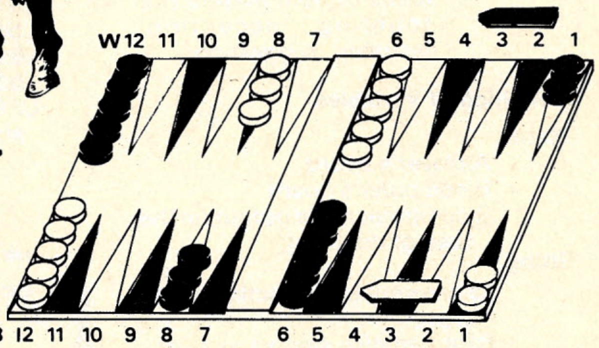
Availability

The Peachtree Word Processing System, on 8-inch diskette with all documentation is $1,500 to the end user. The Computer Systemcenter is at 3330 Peachtree Road, Atlanta, Georgia 30326, phone (404) 231-2308. ■

Another new game from Creative Computing....



GAMMON



Paul von Autenried

Program Description***

GAMMON was designed, written, and completed at Park Ridge High School on a timeshared BASIC system. The computer is a Hewlett Packard System 2000 F, situated at Riverdell High School, in New Jersey.

GAMMON is a program that accurately simulates the ancient game of backgammon. The game itself is condusive to all levels of game play, because of the element of chance.

The program has an advantage over the player in that it generally never "misses" a move. However, it has an absolute degree of consistency which can be gauged by the opponent. It is important to note that the series of move searches was developed by a preference of the programmer. GAMMON was designed with subprogram move searches, so that another programmer can arrange his priorities as he wills.

GAMMON uses two separate matrix lists, one for the player, and one for the computer.

A large portion of GAMMON is devoted to checking that the opponent player does not make a mistake. After all checks have been made, the player's move is recorded. This is accomplished by a series of decision statements followed by the matrix list definitions. The computer move is determined by sections of priority move searches. Generally, the order of the move searches are:

(1) move two men together to form a blocked point
(2) hit the other player
(3) move safely (to a previously blocked point)
(4) move the farthest man.

Decisions of move possibilities within the searches are made by the manipulation of the two basic lists. After each move, a subroutine check is made as to the current status of each player (that is, if either one is completely in his inner table, etc.). The final game play of bearing off is programmed separately.

Since the ability to bet in backgammon, through the use of a doubling cube, may be the most intriguing element of the game, GAMMON has the ability to accept or refuse a double, and suggest a double to its opponent. The program permanently stores all wins and losses with every opponent, who is assigned a user number, in a file named DEBT. The player may obtain the status of the cube by entering a 1 within five seconds after the printout of the board. Entering a 2 will be understood to be suggesting a double, in which case the computer decides if he will accept or decline. Using the same status evaluation the program checks the possibility of promoting a double.

When either player has won, the status of the other player is checked, to determine if he has been gammoned or backgammoned.

GAMMON also has the capability to teach the game of backgammon, with instructions printed by a sister program, INSGAM.

Program Structure

Lines	
100 - 170	Variable and Matrix Initialization
180 - 390	Instruction Printout
400 - 420	Defining board matrices for initial set-up
430 - 590	User Number Search (determining past debts/assigning new user number)
600 - 670	Initial Roll to determine who starts
680 - 750	Player Dice Roll and Printout
760 - 1240	Player Move Input (checking and definition)
1250 - 1410	Computer Dice Roll, printout, and initialization
1420 - 1640	Move Search Routine for forming a new blocked point
1650 - 2440	Move Search Routine for hitting player opponent
2450 - 2530	Computer Move Printout
2540 - 2910	Board Status Printout
2920 - 3020	Player Cube Doubling
3030 - 3600	Move Search Routine for moving without forming a blot

Paul von Autenried, 1 Lockwood Pl., Park Ridge, NJ 07656.

3610 - 4190: Player Move Input (in case of doubles)
4200 - 5120: Computer Move Determination (in case of doubles)
5130 - 5160: Computer Move Printout (in case of doubles)
5170 - 5620: Move Search Routine for moving farthest man
5630 - 6090: Move Routine for coming off the bar
6100 - 6250: Subroutine to check status of computer
6260 - 6430: Game Finish if Computer won
6440 - 6580: Subroutine to check status of player
6590 - 6750: Game Finish if Player won
6760 - 7450: Bearing off Move Search Routine
7460 - 7730: Subroutine for check status of both players (for doubling, accepting, declining, suggesting)

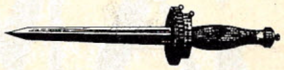
Definition of Variables

Matrix
- A:player's board
- B:computer's board
- S:computer move output (in the case of doubles)

String
- A$:answers to instruction questions
- B$:a negative sign so as to accommodate signs in the printout

Variable
- A:flag variable
- C1:computer's first dice roll
- C2:computer's second dice roll
- C3:computer dice roll (in the case of doubles)
- C4:computer dice roll (in the case of doubles)
- C7:computer's first dice roll (kept constant)
- C8:computer's second dice roll (kept constant)
- D1:player's first dice roll
- D2:player's second dice roll
- F:number of moves remaining (in the case of doubles) to be decremented
- F1:number of moves remaining (in the case of doubles) kept constant
- F2:limit of matrix for printout
- F3:counter of moves for move output definitions
- G:number of computer's chips on the bar (to be decremented)
- G1:number of computer's chips on the bar (kept constant)
- G7:summing variable of number of player chips on various sections of board: status
- G8:summing variable of number of computer chips on various sections of board: status
- H:secondary searching loop of board (in the case of doubles)
- H1:number of player's chips on the bar (to be decremented)
- H2:loop counter for printout of matrix
- H3:variable indicating increment or decrement of dice (in the case of bearing off, if the value cannot be exactly used)
- I:searching loop of board (to determine priorities)
- I1:number of chips on any one point being searched (to determine priorities)
- I2:number of chips on any one point being searched (to determine priorities)
- J:searching loop to determine board status
- L:flag variable
- L5:limit of searching loop of board
- L6:flag variable (computer board status)
- L7:flag variable (player board status)
- M1:player move input (from point #) with first die
- M2:player move input (to point #) with first die
- M3:player move input (from point #) with second die (in the case of doubles)
- M4:player move input (to point #) with second die (in the case of doubles)
- M5:player move input (from point #) with third die (in the case of doubles)
- M6:player move input (to point #) with third die (in the case of doubles)
- M7:player move input (from point #) with fourth die (in the case of doubles)
- M8:player move input (to point #) with fourth die (in the case of doubles)
- N1:player move input (from point #) with second die
- N2:player move input (to point #) with second die

Sample Run



```
ANY INSTRUCTIONS ? YES

BACKGAMMON :     MY CHIPS  ARE REPRESENTED BY - SIGNS, YOUR
   CHIPS BY + SIGNS.  INPUT YOUR MOVES IN THE FORMAT FROM
   POINT #, TO POINT #, FROM POINT #, TO POINT # ; FOR EACH
   MOVE.  EXAMPLE:  5,7,12,17 WOULD BE THE MOVE FOR DICE ROLLS
   OF 2 AND 5 ; MOVING ONE CHIP FROM POINT 5 TO POINT 7, AND
   ANOTHER FROM POINT 12 TO POINT 17.
   THE BAR IS REPRESENTED BY 0, YOUR HOME IS 25, MY
   HOME IS -1.
IF, AT ANY TIME, YOU CARE TO DOUBLE, YOU WILL
BE ALLOWED 5 SECONDS IMMEDIATELY AFTER THE BOARD
PRINOUT TO INPUT A1, INDICATING YOU WISH TO CHECK
WHAT VALUE THE CUBE HAS, OR A 2, SUGGESTING DOUBLING THE
STAKES.

 RULES FOR THE GAME OF BACKGAMMON ? NO
HAVE YOU PLAYED WITH ME BEFORE? NO
YOUR USER NUMBER IS  2

 24 23 22 21 20 19    18 17 16 15 14 13
 .......................................
 -2  0  0  0  0 +5     0 +3  0  0  0 -5
                -0 +0
 +2  0  0  0  0 -5     0 -3  0  0  0 +5
 .......................................
  1  2  3  4  5  6     7  8  9 10 11 12
```

—Game opening—

```
WHAT IS YOUR USER NUMBER ? 2
YOU OWE ME  0    CENTS.  OBVIOUSLY YOU LIKE TO MOVE DEEPER INTO DEBT.
YOUR ROLL IS A  2    AND MY ROLL IS A  3    .
MY ROLL IS  2    AND  3
MY MOVE IS  13    , 11   AND 11    , 8
STATUS :

 24 23 22 21 20 19    18 17 16 15 14 13
 .......................................
 -2  0  0  0  0 +5     0 +3  0  0  0 -4
                -0 +0
 +2  0  0  0  0 -5     0 -4  0  0  0 +5
 .......................................
  1  2  3  4  5  6     7  8  9 10 11 12


YOUR ROLL IS 5    AND 5     .  WHAT IS YOUR MOVE? 12,17,12,17,17,22,17,22
YOUR MOVE IS COMPLETED.
MY ROLL IS  5    AND  4
MY MOVE IS  13    , 8    AND 8     , 4
STATUS :

 24 23 22 21 20 19    18 17 16 15 14 13
 .......................................
 -2  0 +2  0  0 +5     0 +3  0  0  0 -3
                -0 +0
 +2  0  0 -1  0 -5     0 -4  0  C  0 +3
 .......................................
  1  2  3  4  5  6     7  8  9 10 11 12
```

P:limit of matrix for printout
S1:computer move output (from point #) with first die
S2:computer move output (to point #) with first die
T1:computer move output (from point #) with second die
T2:computer move output (to point #) with second die
V:determining in which direction to printout matrix
X:player initial die roll
X1:number of chips on any one point being searched (to determine priorities)
X3:flag variable
X4:limit to searching of board (in the case of doubles)
X5:flag variable
Y:computer initial die roll
Z:switching variable
Z1:switching variable
Z2:switching variable
Z3:switching variable
Z4:switching variable
Z5:switching variable
Z6:switching variable

Variable List Update

File

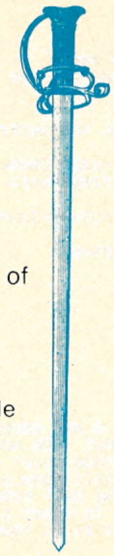
DEBT:permanent record of debts

String

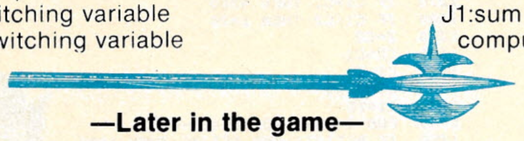
M$:contains losing status

Variable

D4:past debt as read from file
D5:enter statement value
D7:user number
D8:number of users as read from file
D9:doubling cube value
J1:sum of player's chips in computer's inner table



J2:sum of computer's chips in player's inner table
R1:enter statement reference value
R2:loop counter to analyze status on all parts of board for possible doubling
R3:read data in doubles evaluation
R4:searching loop from/to values read from data
R5:read data in doubles evaluation
R6:read data in doubles evaluation
R7:number of computer's chips in analyzed section
R8:number of player's chips in analyzed section
U1:number of moves for computer to win
U2:number of moves for player to win



—Later in the game—

```
YOUR MOVE IS COMPLETED.
MY ROLL IS  5     AND  5
MY MOVE IS 16   , 11   AND 16   , 11   AND 11   , 6   AND 11
STATUS :                                                 , 6

 24 23 22 21 20 19    18 17 16 15 14 13
 .....................................
  0  0 +3 +2 +3 +3     0 +2  0  0  0  0
                -0 +0
 -2 -1 -4  0  0 -7     0 -1  0  0  0 +2
 .....................................
  1  2  3  4  5  6     7  8  9 10 11 12
```

```
YOUR ROLL IS 1     AND 5     .  WHAT IS YOUR MOVE? 12,17,17,18
YOUR MOVE IS COMPLETED.
MY ROLL IS  2     AND  5
MY MOVE IS  8     , 6     AND 6     , 1
STATUS :

 24 23 22 21 20 19    18 17 16 15 14 13
 .....................................
  0  0 +3 +2 +3 +3    +1 +2  0  0  0  0
                -0 +0
 -3 -1 -4  0  0 -7     0  0  0  0  0 +1
 .....................................
  1  2  3  4  5  6     7  8  9 10 11 12
```

```
YOUR ROLL IS 4     AND 2     .  WHAT IS YOUR MOVE? 18,20,17,23
NICE TRY.  NEXT TIME I WILL LOAD THE DICE!  REINPUT!!!!? 18,20,
YOUR MOVE IS COMPLETED.                                  17,21
MY ROLL IS  3     AND  2
MY MOVE IS  3     ,-1     AND 2     ,-1
STATUS :

 24 23 22 21 20 19    18 17 16 15 14 13
 .....................................
  0  0 +3 +3 +4 +3     0 +1  0  0  0  0
                -0 +0
 -3  0 -3  0  0 -7     0  0  0  0  0 +1
 .....................................
  1  2  3  4  5  6     7  8  9 10 11 12
```

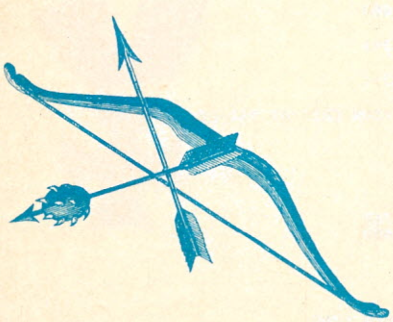
—At the end of the game—

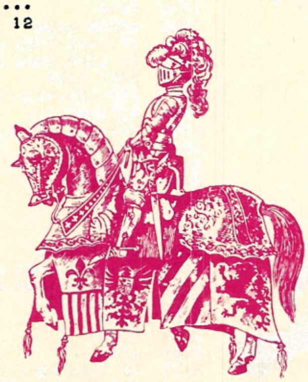
```
YOUR ROLL IS 2     AND 5     .  WHAT IS YOUR MOVE? 20,25,22,24
YOUR MOVE IS COMPLETED.
MY ROLL IS  4     AND  6
MY MOVE IS  3     ,-1     AND 6     ,-1
STATUS :

 24 23 22 21 20 19    18 17 16 15 14 13
 .....................................
 +1  0 +2 +3 +1  0     0  0  0  0  0  0
                -0 +0
  3 -1 -2  0  0  0     0  0  0  0  0  0
 .....................................
  1  2  3  4  5  6     7  8  9 10 11 12
```

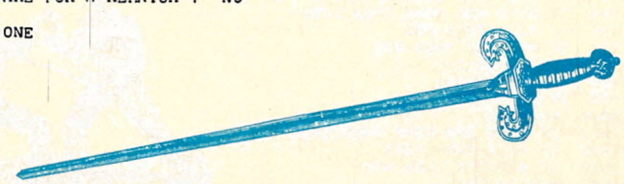
```
YOUR ROLL IS 5     AND 4     .  WHAT IS YOUR MOVE? 20,25,21,25
YOUR MOVE IS COMPLETED.
MY ROLL IS  2     AND  3
MY MOVE IS  2     ,-1     AND 3     ,-1
STATUS :

 24 23 22 21 20 19    18 17 16 15 14 13
 .....................................
 +1  0 +2 +2  0  0     0  0  0  0  0  0
                -0 +0
 -3  0 -1  0  0  0     0  0  0  0  0  0
 .....................................
  1  2  3  4  5  6     7  8  9 10 11 12
```

```
YOUR ROLL IS 5     AND 1     .  WHAT IS YOUR MOVE? 21,25,24,25
YOUR MOVE IS COMPLETED.
MY ROLL IS  1     AND  1
MY MOVE IS 3     , 2     AND 1     ,-1     AND 1     ,-1     AND 1
STATUS :                                                      ,-1

 24 23 22 21 20 19    18 17 16 15 14 13
 .....................................
  0  0 +2 +1  0  0     0  0  0  0  0  0
                -0 +0
  0 -1  0  0  0  0     0  0  0  0  0  0
 .....................................
  1  2  3  4  5  6     7  8  9 10 11 12
```

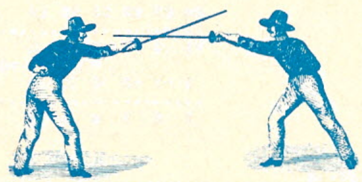
```
YOUR ROLL IS 1     AND 5     .  WHAT IS YOUR MOVE? 21,25,22,23
YOUR MOVE IS COMPLETED.
MY ROLL IS  1     AND  6
I WON.
YOU HAVE LOST 4 POINTS.  I'LL REMEMBER THAT.
CARE FOR A REMATCH ?  NO

DONE
```





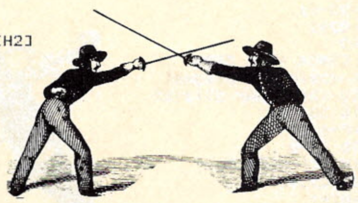


Program Listing

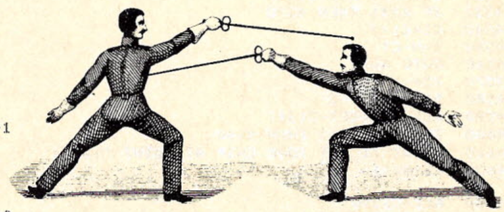


```
10  REM                  ***** GAMMON *****
20  REM        **** A SIMULATION OF THE GAME OF BACKGAMMON ****
30  REM
40  REM
50  REM            *** WRITTEN BY PAUL VON AUTENRIED ***
60  REM
70  REM              ** AT PARK RIDGE HIGH SCHOOL **
80  REM         * ON HEWLETT PACKARD TIMESHARED SYSTEM 2000F *
90  REM
100  COM A[25],B[25],S[10],A$[3],M$[13],P$[3],R$[3]
110  FILES DEBT
120  L7=L6=X5=L=S1=S2=T1=T2=H1=G=P2=D5=0
130  D9=1
140  B$="-"
150  MAT A=ZER
160  MAT B=ZER
170  MAT S=ZER
180  PRINT "ANY INSTRUCTIONS ";
190  INPUT A$
200  IF A$[1,1]="N" THEN 400
210  PRINT
220  PRINT "BACKGAMMON :      MY CHIP S ARE REPRESENTED BY - SIGNS, YOUR"
230  PRINT "   CHIPS BY + SIGNS.  INPUT YOUR MOVES IN THE FORMAT FROM"
240  PRINT "   POINT #, TO POINT #, FROM POINT #, TO POINT # ; FOR EACH "
250  PRINT "   MOVE.  EXAMPLE:  5,7,12,17 WOULD BE THE MOVE FOR DICE ROLLS"
260  PRINT "   OF 2 AND 5 ; MOVING ONE CHIP FROM POINT 5 TO POINT 7, AND"
270  PRINT "   ANOTHER FROM POINT 12 TO POINT 17."
280  PRINT "   THE BAR IS REPRESENTED BY 0, YOUR HOME IS 25, MY"
290  PRINT "   HOME IS 0."
300  PRINT "IF, AT ANY TIME, YOU CARE TO DOUBLE, YOU WILL"
310  PRINT "BE ALLOWED 5 SECONDS IMMEDIATELY AFTER THE BOARD"
320  PRINT "PRINOUT TO INPUT A1, INDICATING YOU WISH TO CHECK"
330  PRINT "WHAT VALUE THE CUBE HAS, OR A 2, SUGGESTING DOUBLING THE"
340  PRINT "STAKES."
350  PRINT
360  PRINT " RULES FOR THE GAME OF BACKGAMMON ";
370  INPUT A$
380  IF A$[1,1]="N" THEN 400
390  CHAIN "INSGAM"
400  A[19]=B[6]=A[12]=B[13]=5
410  A[1]=B[24]=2
420  A[17]=B[8]=3
430  PRINT "HAVE YOU PLAYED WITH ME BEFORE";
440  INPUT P$
450  IF P$[1,1]="Y" THEN 520
460  READ #1,1
470  READ #1;D8
480  D8=D8+1
490  PRINT "YOUR USER NUMBER IS ";D8+1
500  PRINT #1,1;D8
510  D7=D8
520  GOSUB 2570
530  PRINT "WHAT IS YOUR USER NUMBER ";
540  INPUT D7
550  READ #1,D7;D4
560  IF D4<0 THEN 590
570  PRINT "YOU OWE ME ";D4;"CENTS.  OBVIOUSLY YOU LIKE TO MOVE DEEPER INTO DEBT."
580  GOTO 600
590  PRINT "I OWE YOU ";-D4;"CENTS.  NICE OF YOU TO RETURN."
600  X=INT((RND(0)*6)+1)
610  Y=INT((RND(0)*6)+1)
620  PRINT "YOUR ROLL IS A ";X;"AND MY ROLL IS A ";Y;"."
630  IF X#Y THEN 670
640  D9=D9*2
650  PRINT "THE DOUBLING CUBE IS NOW AT ";D9
660  GOTO 600
670  IF Y>X THEN 1260
680  PRINT "YOU START."
690  D1=X
700  D2=Y
710  GOTO 740
720  D1=INT((RND(0)*6)+1)
730  D2=INT((RND(0)*6)+1)
740  IF H1>0 AND B[D1]>1 AND B[D2]>1 THEN 1220
750  PRINT "YOUR ROLL IS";D1;"AND";D2;".  WHAT IS YOUR MOVE";
760  GOSUB 6440
770  IF D1=D2 THEN 3610
780  INPUT M1,M2,N1,N2
790  IF L7=1 THEN 960
800  IF M1=0 AND M2=0 THEN 850
810  IF M2-M1=D1 THEN 850
820  IF M2-M1=D2 THEN 880
830  PRINT "NICE TRY.  NEXT TIME I WILL LOAD THE DICE!  REINPUT!!!!";
840  GOTO 780
850  IF N1=0 AND N2=0 THEN 930
860  IF N2-N1#D2 THEN 830
870  GOTO 900
880  IF N1=0 AND N2=0 THEN 930
890  IF N2-N1#D1 THEN 830
900  IF B[M2]>1 OR B[N2]>1 THEN 830
910  IF M1=0 AND M2=0 THEN 940
920  IF N1=0 AND N2=0 THEN 940
930  IF M1#0 AND N1#0 AND H1>0 THEN 830
940  IF M1=0 AND M2=0 THEN 1010
950  IF M1<1 THEN 980
960  IF A[M1]<1 THEN 830
970  A[M1]=A[M1]-1
980  A[M2]=A[M2]+1
990  IF M1>0 THEN 1010
1000  H1=H1-1
1010  IF N1=0 AND N2=0 THEN 1080
1020  IF N1<1 THEN 1050
1030  IF A[N1]<1 THEN 830
1040  A[N1]=A[N1]-1
1050  A[N2]=A[N2]+1
1060  IF N1>0 THEN 1080
1070  H1=H1-1
1080  IF M1=0 AND M2=0 THEN 1140
1090  IF B[M2]<1 THEN 1140
1100  B[M2]=B[M2]-1
1110  G=G+1
1120  PRINT "YOU HIT ME !!!!"
1130  IF M2=N2 THEN 1190
1140  IF N2=0 AND N1=0 THEN 1190
1150  IF B[N2]<1 THEN 1190
1160  B[N2]=B[N2]-1
1170  G=G+1
1180  PRINT "YOU HIT ME !!!!"
1190  PRINT "YOUR MOVE IS COMPLETED."
1200  GOSUB 7460
1210  GOTO 1230
1220  PRINT "YOUR ROLL IS ";D1;"AND ";D2;" BUT YOU CAN NOT MOVE HA-HA!!!!"
1230  GOSUB 6440
1240  GOTO 1290
1250  REM COMPUTER MOVE DETERMINATION
1260  C1=X
1270  C2=Y
1280  GOTO 1330
1290  S1=S2=T1=T2=0
1300  MAT S=ZER
1310  C1=INT((RND(0)*6)+1)
1320  C2=INT((RND(0)*6)+1)
1330  GOSUB 6100
1340  PRINT "MY ROLL IS ";C1;"AND ";C2
1350  IF G>0 THEN 5630
1360  IF L6=1 THEN 6760
1370  IF C1=C2 THEN 4200
1380  IF C1<C2 THEN 1420
1390  Z=C2
1400  C2=C1
1410  C1=Z
1420  L5=C2+1
1430  I1=15
1440  FOR I=C1+1 TO 24-(C2-C1)
1450  IF B[I]<I1 AND B[I]#1 THEN 1480
1460  IF B[I-C1]>1 THEN 1480
1470  IF A[I-C1]<2 THEN 1530
1480  NEXT I
1490  I1=I1-1
1500  IF I1=2 THEN 1650
1510  IF I1=0 THEN 5170
1520  GOTO 1440
1530  IF B[I+(C2-C1)]<1 THEN 1480
1540  IF A[I-C1]=0 THEN 1580
1550  PRINT "I GOT YOU."
1560  A[I-C1]=A[I-C1]-1
1570  H1=H1+1
1580  B[I]=B[I]-1
1590  B[I-C1]=B[I-C1]+2
1600  B[I+(C2-C1)]=B[I+(C2-C1)]-1
1610  S1=I
1620  S2=T2=I-C1
1630  T1=I+(C2-C1)
1640  GOTO 2450
1650  L5=13
1660  FOR I=24 TO L5 STEP -1
1670  IF B[I]<1 THEN 1780
1680  IF C1=0 THEN 1700
1690  IF A[I-C1]=1 THEN 1840
1700  IF C2=0 THEN 1780
1710  IF A[I-C2]=1 THEN 1970
1720  IF I-C1-C2<1 THEN 1740
1730  IF A[I-C1-C2]#1 THEN 1780
1740  IF A[I-C1]>1 THEN 1760
1750  GOTO 2130
1760  IF A[I-C2]>1 THEN 1780
1770  GOTO 2100
1780  NEXT I
1790  IF L5#13 OR C2=0 THEN 1810
1800  L5=C2+1
1810  IF I1=2 THEN 3030
1820  IF I1=1 AND C1#0 AND C2#0 THEN 1440
1830  GOTO 5170
1840  B[I-C1]=B[I-C1]+1
1850  B[I]=B[I]-1
1860  A[I-C1]=A[I-C1]-1
1870  H1=H1+1
1880  PRINT "I TOOK YOUR CHIP AT ";I-C1
1890  L5=C2+1
1900  S1=I
1910  S2=I-C1
1920  C1=0
1930  GOSUB 6100
1940  IF L6=1 THEN 6760
1950  IF C2=0 THEN 2450
1960  GOTO 1670
1970  B[I-C2]=B[I-C2]+1
1980  B[I]=B[I]-1
1990  A[I-C2]=A[I-C2]-1
2000  H1=H1+1
2010  PRINT "I TOOK YOUR CHIP AT ";I-C2
2020  L5=C1+1
2030  T1=I
2040  T2=I-C2
2050  C2=0
2060  GOSUB 6100
2070  IF L6=1 THEN 6760
2080  IF C1=0 THEN 2450
2090  GOTO 1670
2100  Z=C2
2110  C2=C1
2120  C1=Z
2130  IF I-C1-C2<1 THEN 1780
2140  B[I-C1-C2]=B[I-C1-C2]+1
```

```
2150  B[I]=B[I]-1
2160  A[I-C1-C2]=A[I-C1-C2]-1
2170  H1=H1+1
2180  S1=I
2190  S2=T1=I-C1
2200  T2=I-C1-C2
2210  PRINT "I TOOK YOUR CHIP AT ";I-C1-C2
2220  C1=C2=0
2230  IF C1=0 AND C2=0 THEN 2450
2240  IF C1#0 AND C2#0 THEN 3030
2250  IF C2=0 THEN 2290
2260  Z=C2
2270  C2=C1
2280  C1=Z
2290  FOR I=C1+1 TO 24
2300  IF B[I]<1 THEN 2320
2310  IF B[I-C1]>0 THEN 2370
2320  NEXT I
2330  Z=C2
2340  C2=C1
2350  C1=Z
2360  GOTO 3030
2370  B[I]=B[I]-1
2380  B[I-C1]=B[I-C1]+1
2390  IF S1=0 THEN 2430
2400  T1=I
2410  T2=I-C1
2420  GOTO 2450
2430  S1=I
2440  S2=I-C1
2450  REM COMPUTER MOVE PRINTOUT
2460  IF S1#T2 THEN 2530
2470  Z1=S1
2480  Z2=S2
2490  S1=T1
2500  S2=T2
2510  T1=Z1
2520  T2=Z2
2530  PRINT "MY MOVE IS ";S1;",";S2;"AND";T1;",";T2
2540  PRINT "STATUS :"
2550  GOSUB 2570
2560  GOTO 720
2570  REM PRINTOUT
2580  F2=24
2590  P=13
2600  V=-1
2610  PRINT
2620  REM
2630  PRINT " 24 23 22 21 20 19   18 17 16 15 14 13"
2640  PRINT " ....................................."
2650  PRINT TAB(1);
2660  FOR H2=F2 TO P STEP V
2670  IF A[H2]<1 THEN 2710
2680  PRINT  USING 2690;A[H2]
2690  IMAGE #,SDX
2700  GOTO 2760
2710  IF B[H2]<1 THEN 2740
2720  PRINT  USING 2690;-B[H2]
2730  GOTO 2760
2740  PRINT  USING 2750;0
2750  IMAGE #,XDX
2760  IF H2=19 THEN 2790
2770  IF H2=6 THEN 2790
2780  GOTO 2800
2790  PRINT TAB(21);
2800  NEXT H2
2810  IF H2=13 THEN 2890
2820  PRINT
2830  PRINT  USING 2840;TAB(16),B$,G,H1
2840  IMAGE X,A,DX,SD
2850  V=1
2860  P=12
2870  F2=1
2880  GOTO 2650

2890  PRINT " ....................................."
2900  PRINT "  1  2  3  4  5  6    7  8  9 10 11 12"
2910  PRINT LIN(-2)
2920  ENTER 5,R1,D5
2930  GOTO D5+1 OF 3010,2940,7460
2940  PRINT "THE DOUBLING CUBE IS AT ";
2950  IF D9<0 THEN 2990
2960  PRINT D9;".  YOU MAY DOUBLE IF YOU WISH."
2970  D5=0
2980  GOTO 2920
2990  PRINT -D9;"IT IS MY TURN TO DOUBLE."
3000  D5=0
3010  GOSUB 6100
3020  RETURN
3030  REM SAFE MOVE
3040  FOR I=24 TO L5 STEP -1
3050  IF B[I]<1 THEN 3190
3060  IF C1=0 OR C2=0 THEN 3100
3070  IF I-C1-C2<1 THEN 3100
3080  IF B[I-C1-C2]>0 AND A[I-C1]<1 THEN 3250
3090  IF B[I-C1-C2]>0 AND A[I-C2]<1 THEN 3450
3100  IF C1=0 THEN 3130
3110  IF B[I-C1]>0 THEN 3220
3120  IF C2=0 THEN 3190
3130  IF B[I-C2]>0 THEN 3420
3140  IF A[I-C1]>0 THEN 3180
3150  IF I-C1-C2<1 THEN 3190
3160  IF B[I-C1-C2]<1 THEN 3190
3170  GOTO 3250
3180  IF A[I-C2]<1 AND B[I-C1-C2]>1 THEN 3450
3190  NEXT I
3200  I1=I1-1
3210  GOTO 1660
3220  IF C2=0 THEN 3310
```

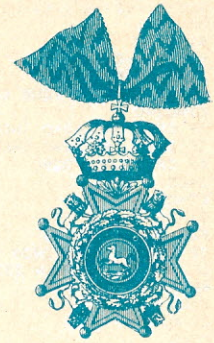


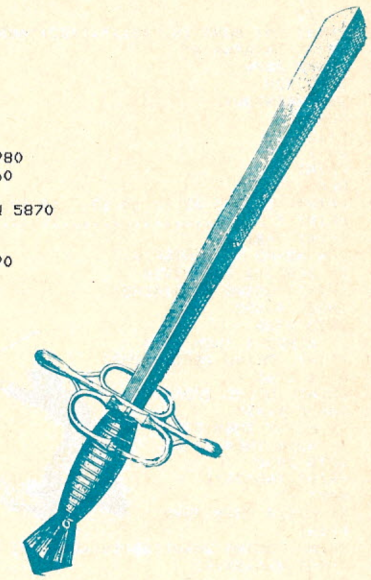
```
3230  IF I-C1-C2<1 THEN 3190
3240  IF B[I-C1-C2]<1 THEN 3310
3250  B[I-C1-C2]=B[I-C1-C2]+1
3260  B[I]=B[I]-1
3270  S1=I
3280  S2=T1=I-C1
3290  T2=I-C1-C2
3300  GOTO 2450
3310  B[I]=B[I]-1
3320  B[I-C1]=B[I-C1]+1
3330  S1=I
3340  S2=I-C1
3350  C1=0
3360  GOSUB 6100
3370  IF L6=1 THEN 6760
3380  L5=C2+1
3390  IF C2=0 THEN 2450
3400  IF B[I]>0 THEN 3130
3410  GOTO 3190
3420  IF C1=0 THEN 3510
3430  IF I-C1-C2<1 THEN 3510
3440  IF B[I-C1-C2]<1 THEN 3510
3450  B[I-C1-C2]=B[I-C1-C2]+1
3460  B[I]=B[I]-1

3470  S1=I
3480  S2=T1=I-C2
3490  T2=I-C1-C2
3500  GOTO 2450
3510  B[I]=B[I]-1
3520  B[I-C2]=B[I-C2]+1
3530  T1=I
3540  T2=I-C2
3550  C2=0
3560  GOSUB 6100
3570  IF L6=1 THEN 6760
3580  L5=C1+1
3590  IF C1=0 THEN 2450
3600  GOTO 3190
3610  REM PLAYER DOUBLES
3620  INPUT M1,M2,M3,M4,M5,M6,M7,M8
3630  IF L7=1 THEN 3750
3640  IF M1=0 AND M2=0 THEN 3680
3650  IF M2-M1=D1 THEN 3680
3660  PRINT "NICE TRY.  NEXT TIME, I WILL LOAD THE DICE.   REINPUT !!!!";
3670  GOTO 3620
3680  IF M3=0 AND M4=0 THEN 3700
3690  IF M4-M3#D1 THEN 3660
3700  IF M5=0 AND M6=0 THEN 3720
3710  IF M6-M5#D1 THEN 3660
3720  IF M7=0 AND M8=0 THEN 3740
3730  IF M8-M7#D1 THEN 3660
3740  IF B[M2]>1 OR B[M4]>1 OR B[M6]>1 OR B[M8]>1 THEN 3660
3750  IF M1=0 AND M2=0 THEN 3820
3760  IF M1<1 THEN 3790
3770  IF A[M1]<1 THEN 3660
3780  A[M1]=A[M1]-1
3790  A[M2]=A[M2]+1
3800  IF M1>0 THEN 3820
3810  H1=H1-1
3820  IF M3=0 AND M4=0 THEN 3890
3830  IF M3<1 THEN 3860
3840  IF A[M3]<1 THEN 3660
3850  A[M3]=A[M3]-1
3860  A[M4]=A[M4]+1
3870  IF M3>0 THEN 3890
3880  H1=H1-1
3890  IF M5=0 AND M6=0 THEN 3960
3900  IF M5<1 THEN 3930
3910  IF A[M5]<1 THEN 3660
3920  A[M5]=A[M5]-1
3930  A[M6]=A[M6]+1
3940  IF M5>0 THEN 3960
3950  H1=H1-1
3960  IF M7=0 AND M8=0 THEN 4030
3970  IF M7<1 THEN 4000
3980  IF A[M7]<1 THEN 3660
3990  A[M7]=A[M7]-1
4000  A[M8]=A[M8]+1
4010  IF M7>0 THEN 4030
4020  H1=H1-1
4030  IF B[M2]#1 THEN 4070
4040  B[M2]=B[M2]-1

4050  G=G+1
4060  PRINT "YOU HIT ME !!!!"
4070  IF B[M4]#1 THEN 4110
4080  B[M4]=B[M4]-1
4090  G=G+1
4100  PRINT "YOU HIT ME !!!!"
4110  IF B[M6]#1 THEN 4150
4120  B[M6]=B[M6]-1
4130  G=G+1
4140  PRINT "YOU HIT ME !!!!"
4150  IF B[M8]#1 THEN 4190
4160  B[M8]=B[M8]-1
4170  G=G+1
4180  PRINT "YOU HIT ME !!!!"
4190  GOTO 1190
4200  REM COMPUTER DOUBLES
4210  X3=L=0
4220  MAT S=ZER
4230  C3=C4=C1
4240  L5=C1+1
4250  FOR I=L5 TO 24
4260  IF B[I]<2 THEN 4750
4270  IF A[I-C1]>1 THEN 4750
4280  IF I-(2*C1)<1 THEN 4340
```

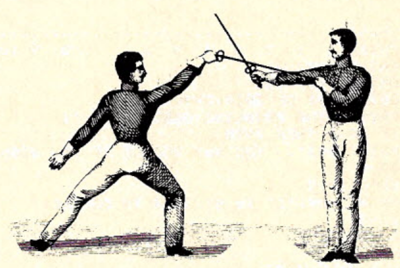


```
4290  IF A[I-(2*C1)]#1 THEN 4340
4300  C1=2*C1
4310  H=I
4320  X3=1
4330  GOTO 4420
4340  IF B[I-C1]>1 THEN 4770
4350  IF B[I]-2=1 THEN 4770
4360  X4=I+1
4370  FOR H=24 TO X4 STEP -1
4380  X5=0
4390  IF B[H]<2 THEN 4700
4400  IF A[H-C1]>1 THEN 4700
4410  IF A[H-C1]=0 THEN 4450
4420  PRINT "I HIT YOUR CHIP AT POINT ";H-C1
4430  H1=H1+1
4440  A[H-C1]=0
4450  IF X3#1 THEN 4470
4460  C1=C1/2
4470  S[2]=S[4]=H-C1
4480  S[1]=S[3]=H
4490  B[H]=B[H]-2
4500  IF X3#1 THEN 4520
4510  C1=C2*2
4520  B[H-C1]=B[H-C1]+2
4530  IF X3#1 THEN 4570
4540  C1=C1/2
4550  I=H-C1
4560  GOTO 4640
4570  X5=-1
4580  B[I]=B[I]-2
4590  B[I-C1]=B[I-C1]+2
4600  IF A[I-C1]=0 THEN 4640
4610  PRINT "I HIT YOUR CHIP AT POINT ";I-C1
4620  H1=H1+1
4630  A[I-C1]=0
4640  S[5]=S[7]=I
4650  S[6]=S[8]=I-C1
4660  IF X5#-1 THEN 4680
4670  C1=C2=C3=C4=0
4680  C1=C2=0
4690  GOTO 4800
4700  NEXT H
4710  IF X4#I+1 THEN 4580
4720  X4=C4+1
4730  GOTO 4700
4740  GOTO 4580
4750  NEXT I
4760  GOTO 4800
4770  IF L>0 THEN 4360
4780  L=L+1
4790  GOTO 4750
4800  REM COMPUTER DOUBLES SAFE MOVE
4810  IF C1=0 AND C2=0 AND C3=0 AND C4=0 THEN 5030
4820  IF C1#0 AND C2#0 AND C3#0 AND C4#0 THEN 4850
4830  F1=F=2
4840  GOTO 4870
4850  F1=F=4
4860  L5=C4+1
4870  FOR I=24 TO L5 STEP -1
4880  IF B[I]<1 THEN 5020
4890  IF A[I-C4]>1 THEN 5020
4900  IF A[I-C4]=0 THEN 4940
4910  PRINT "I HIT YOUR CHIP AT POINT ";I-C4
4920  H1=H1+1
4930  A[I-C4]=0
4940  B[I-C4]=B[I-C4]+1
4950  B[I]=B[I]-1
4960  S[(2*F)-1]=I
4970  S[2*F]=I-C4
4980  F=F-1
4990  IF F#0 AND B[I]#0 THEN 4940
5000  IF F#0 AND B[I]=0 THEN 5020
5010  GOTO 5030
5020  NEXT I
5030  FOR I=1 TO 5 STEP 2
5040  IF S[I+1] >= S[I+3] THEN 5120
5050  Z3=S[I+2]
5060  Z4=S[I+3]
5070  S[I+2]=S[I]
5080  S[I+3]=S[I+1]
5090  S[I]=Z3
5100  S[I+1]=Z4
5110  GOTO 5030
5120  NEXT I
5130  PRINT "MY MOVE IS";S[1];",";S[2];"AND";S[3];",";S[4];
5140  PRINT "AND";S[5];",";S[6];"AND";S[7];",";S[8]
5150  GOSUB 6100
5160  GOTO 2540
5170  REM COMPUTER FARTHEST MOVE
5180  C9=0
5190  IF C2=0 THEN 5220
5200  L5=C2+1
5210  GOTO 5230
5220  L5=C1+1
5230  X1=1
5240  GOSUB 6100
5250  IF L6=1 THEN 6760
5260  FOR I=24 TO L5 STEP -1
5270  IF B[I]#X1 THEN 5380
5280  IF C2=0 THEN 5370
5290  IF A[I-C2]>1 THEN 5530
5300  B[I]=B[I]-1
5310  B[I-C2]=B[I-C2]+1
5320  T1=I
5330  T2=I-C2
5340  C2=0
5350  L5=C1+1
5360  IF C1=0 THEN 2450
```


```
5370  IF A[I-C1]<1 THEN 5540
5380  NEXT I
5390  IF X1#1 THEN 5410
5400  X1=16
5410  X1=X1-1
5420  IF X1>0 THEN 5260
5430  IF C1=0 OR C9=1 THEN 5470
5440  C9=1
5450  C2=0
5460  GOTO 5220
5470  IF C1#0 THEN 5500
5480  S1=S2=0
5490  GOTO 2450
5500  IF C2#0 THEN 6080
5510  T1=T2=0
5520  GOTO 2450
5530  IF C1=0 OR A[I-C1]>1 THEN 5380
5540  IF B[I]<1 THEN 5380
5550  B[I]=B[I]-1
5560  B[I-C1]=B[I-C1]+1
5570  S1=I
5580  S2=I-C1
5590  C1=0
5600  L5=C2+1
5610  IF C2=0 THEN 2450
5620  GOTO 5270
5630  REM BLOT
5640  I1=1
5650  G1=G
5660  C3=C1
5670  C4=C2
5680  IF A[25-C1]=1 THEN 6000
5690  IF A[25-C2]=1 THEN 6040
5700  IF A[25-C1]<1 AND B[25-C1]>0 THEN 5890
5710  IF A[25-C2]>1 THEN 5870
5720  B[25-C2]=B[25-C2]+1
5730  T1=0
5740  T2=25-C2
5750  L5=C1+1
5760  C2=0
5770  IF C3#C4 THEN 5800
5780  S[9-(2*(G MAX 2))]=T1
5790  S[10-(2*(G MAX 2))]=T2
5800  G=G-1
5810  IF G=0 AND C3=C4 THEN 5980
5820  IF G=0 AND G1=2 THEN 2460
5830  IF G=0 THEN 1660
5840  IF S[7]=0 OR S[5]=0 THEN 5870
5850  C1=C3
5860  C2=C4
5870  IF C1=0 OR C2=0 THEN 5390
5880  IF A[25-C1]>1 THEN 6080
5890  B[25-C1]=B[25-C1]+1
5900  S1=0
5910  S2=25-C1
5920  L5=C2+1
5930  C1=0
5940  IF C3#C4 THEN 5800
5950  S[9-(2*(G MAX 2))]=S1
5960  S[10-(2*(G MAX 2))]=S2
5970  GOTO 5800
5980  F1=F=4-G1
5990  GOTO 4870
6000  PRINT "I HIT YOU."
6010  A[25-C1]=0
6020  H1=H1+1
6030  GOTO 5890
6040  PRINT "I HIT YOU."
6050  A[25-C2]=0
6060  H1=H1+1
6070  GOTO 5720
6080  PRINT "I CAN NOT  MOVE."
6090  GOTO 2540
6100  REM COMPUTER HOME
6110  G8=0
6120  FOR J=7 TO 24
6130  G8=G8+B[J]
6140  NEXT J
6150  IF G8#0 THEN 6240
6160  G8=0
6170  FOR J=1 TO 6
6180  G8=G8+B[J]
6190  NEXT J
6200  IF G8=0 THEN 6260
6210  L5=-1
6220  L6=1
6230  GOTO 6250
6240  L6=0
6250  RETURN
6260  PRINT "I WON."
6270  IF G7#0 AND G8#0 THEN 6380
6280  J1=A[1]+A[2]+A[3]+A[4]+A[5]+A[6]
6290  IF A[7]+A[8]+A[9]+A[10]+A[11]+A[12]+A[13]+A[14]+A[15]+A[16]
      +A(17)+A(18)=0 THEN 6310
6300  M$="GAMMONED."
6310  IF J1=0 THEN 6330
6320  M$="BACKGAMMONED."
6330  IF M$[1,1]#"G" THEN 6350
6340  D9=D9*2
6350  IF M$[1,1]#"B" THEN 6370
6360  D9=D9*3
6370  PRINT "YOU WERE ";M$
6380  PRINT "YOU HAVE LOST ";ABS(D9);"POINTS.  I'LL REMEMBER THAT."
6390  PRINT #1,D7;D4+ABS(D9)
6400  PRINT "CARE FOR A REMATCH ";
6410  INPUT A$
6420  IF A$[1,1]="Y" THEN 100
6430  STOP
```







```
6440  REM PLAYER WIN SUBROUTINE
6450  G7=H1
6460  FOR J=1 TO 18
6470  G7=G7+A[J]
6480  NEXT J
6490  IF G7#0 THEN 6570
6500  G7=0
6510  FOR J=24 TO 19 STEP -1
6520  G7=G7+A[J]
6530  NEXT J
6540  IF G7=0 THEN 6590
6550  L7=1
6560  GOTO 6580
6570  L7=0
6580  RETURN
6590  PRINT "YOU WON."
6600  IF G7#0 AND G8#0 THEN 6710
6610  J2=B[24]+B[23]+B[22]+B[21]+B[20]+B[19]
6620  IF B[18]+B[17]+B[16]+B[15]+B[14]+B[13]+B[12]+B[11]+B[10]+B[9]
      +B(8)+B(7)=0 THEN 6640
6630  M$="GAMMONED."
6640  IF J2=0 THEN 6660
6650  M$="BACKGAMMONED."
6660  IF M$[1,1]#"G" THEN 6680
6670  D9=D9*2
6680  IF M$[1,1]#"B" THEN 6700
6690  D9=D9*3
6700  PRINT "I WAS ";M$
6710  PRINT "I HAVE LOST ";ABS(D9);"POINTS. "
6720  PRINT #1,D7;D4-ABS(D9)
6730  PRINT "GOOD BYE."
6740  PRINT LIN(-5)
6750  STOP
6760  REM COMPUTER FINISH
6770  I2=9
6780  H3=F3=1
6790  C3=C4=0
6800  IF C1#C2 THEN 6830
6810  C3=C1
6820  C4=C2
6830  C7=C1
6840  C8=C2
6850  FOR I=6 TO 1 STEP -1
6860  IF I=C1 AND B[I] >= I2 THEN 7230
6870  IF I=C2 AND B[I] >= I2 THEN 7050
6880  IF I=C3 AND B[I] >= I2 THEN 7360
6890  IF I=C4 AND B[I] >= I2 THEN 7380
6900  NEXT I
6910  I2=I2-1
6920  IF I2#0 THEN 6850
6930  IF C1=0 THEN 6950
6940  C1=C1+H3
6950  IF C2=0 THEN 6970
6960  C2=C2+H3
6970  IF C3=0 THEN 6990
6980  C3=C3+H3
6990  IF C4=0 THEN 7010
7000  C4=C4+H3
7010  IF C1#7 AND C2#7 AND C3#7 AND C4#7 THEN 7030
7020  H3=-1
7030  I2=9
7040  GOTO 6850
7050  S[F3]=T1=I
7060  IF I-C8<1 THEN 7150
7070  IF A[I-C8]>1 THEN 6900
7080  IF A[I-C8]<1 THEN 7120
7090  PRINT "I TOOK YOUR CHIP AT ";I-C8
7100  H1=H1+1
7110  A[I-C8]=A[I-C8]-1
7120  B[I-C8]=B[I-C8]+1
7130  S[F3+1]=T2=I-C8
7140  GOTO 7160
7150  S[F3+1]=T2=-1
7160  C2=0
7170  B[I]=B[I]-1
7180  GOSUB 6100
7190  IF C1=0 AND C2=0 AND C3=0 AND C4=0 AND C7=C8 THEN 5030
7200  IF C1=0 AND C2=0 AND C3=0 AND C4=0 THEN 2460
7210  F3=F3+2
7220  GOTO 6870
7230  S[F3]=S1=I
7240  IF I-C7<1 THEN 7330
7250  IF A[I-C7]>1 THEN 6900
7260  IF A[I-C7]<1 THEN 7300
7270  PRINT "I TOOK YOUR CHIP AT ";I-C7
7280  H1=H1+1
7290  A[I-C7]=A[I-C7]-1
7300  B[I-C7]=B[I-C7]+1
7310  S[F3+1]=S2=I-C7
7320  GOTO 7340
7330  S[F3+1]=S2=-1
7340  C1=0
7350  GOTO 7170
7360  A=1
7370  GOTO 7390
7380  A=2
7390  Z5=C3
7400  C3=C1
7410  C1=Z5
7420  Z6=C4
7430  C4=C2
7440  C2=Z6
7450  GOTO A OF 7230,7050
7460  REM DOUBLES EVALUATION
7470  IF ABS(D9) >= 64 THEN 7720
7480  U1=U2=0
7490  FOR R2=1 TO 5
7500  R7=R8=0
7510  READ R3,R5,R6
7520  FOR R4=R3 TO R5
7530  R7=B[R4]+R7
7540  R8=A[R4]+R8
7550  NEXT R4
7560  U1=U1+(R6*R7)
7570  U2=U2+((3-R6)*R8)
7580  NEXT R2
7590  DATA 1,6,.5,7,12,1,13,15,1.5,16,19,2,20,24,2.5
7600  RESTORE
7610  IF U1 <= U2 AND D9>0 AND D5=2 THEN 7690
7620  IF U1 <= U2 AND D5#2 AND D9<0 THEN 7660
7630  IF D5#2 THEN 7720
7640  PRINT "I REFUSE TO DOUBLE THE  STAKES."
7650  GOTO 6590
7660  PRINT "CARE TO DOUBLE THE STAKES ";
7670  INPUT R$
7680  IF R$[1,1]#"Y" THEN 6260
7690  D9=D9*-2
7700  PRINT "FINE."
7710  D5=0
7720  RETURN
7730  END
```

Answers to Problems on Page 118

Goops, Gorps, and Gorgs: 10 Gorps are neither Goops nor Gorgs.

Square Root, Root, Root, Root: The problem can be solved with a 3-line Basic program, a pocket calculator or algebraically by realizing that the equation to be solved is really:

$$x = \sqrt{12 + x}$$

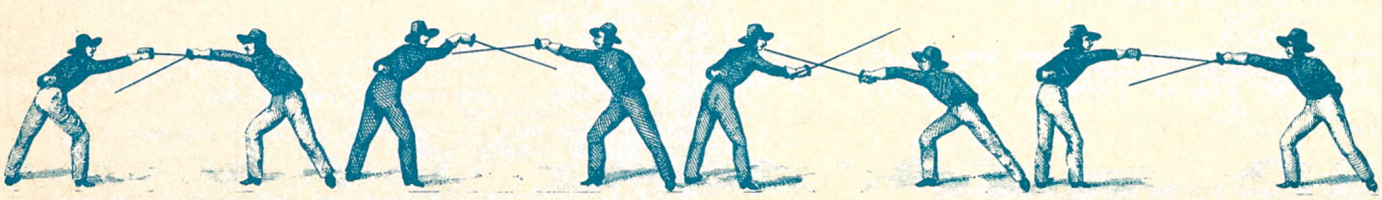
Any way, the solution is 4.

Train and Bee:

One can waste a lot of time on problems that have no solution. We want to realize here that since the bee is flying at a rate faster than the trains are moving, the problem is sensible and has a solution. If one train had a rate faster than that of the bee's, the bee could not fly back and forth between them. (Perhaps you can't perceive of a bee doing this—that's OK—the problem still has a meaningful mathematical solution).

Once we realize that the problem is meaningful, the solution is readily at hand. Since the bee is flying at 33 Km./hr., he will fly 33 kilometers in the last hour before collision.

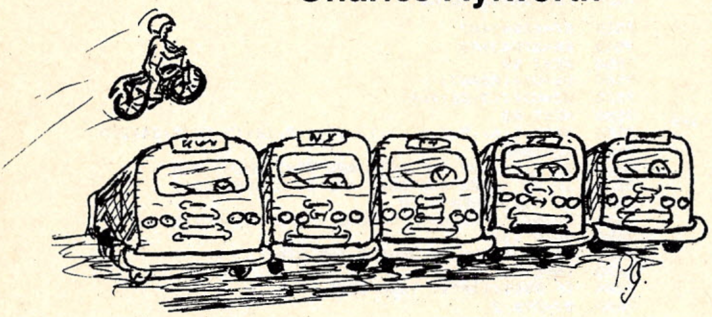
Note that the numerical rates of the trains are irrelevant. We need only know that they are less than that of the bee's rate. This is an example of a problem where one is given more information



Another new game from Creative Computing . . . .

EVILK

Charles Aylworth



EVILK is a program written in North Star BASIC that permits you to act out your fantasies of being a motorcycle daredevil! The game is a simple motorcycle jump over several busses, which takes into account both gravity and drag forces. The ramp angle and motorcycle speed determine the distance jumped. Note that the injury penalty is greater for long jumps than for short ones, and that there is a chance for a crash even on a jump of the right length. This probability, initially set at .20, can be modified in line 560 to make survival more or less likely.

An improved FORTRAN version of EVILK, with more sophisticated calculations of drag forces, will be available from Charles for $.50 and an SASE.

Sample Run

```
WE'RE AT THE SCENE OF THE BIG MOTORCYCLE JUMP!
HOW MANY BUSSES WILL YOU TRY TO JUMP? 5
 5 BUSSES!  THAT'S  75 FEET!
WHAT RAMP ANGLE WILL YOU USE? 90
 90 DEGREES?  THAT'S IMPOSSIBLE.  COME ON NOW,
WHAT RAMP ANGLE WILL YOU USE? 22
HOW FAST WILL YOU LEAVE THE RAMP? 0
      GOOD LUCK!
A PRACTICE JUMP!
OK, THIS TIME HOW FAST WILL YOU LEAVE THE RAMP? 54
      GOOD LUCK!
THERE HE GOES!!!!
******************HE JUMPED TOO FAR!   HE MISSED THE RAMP.
I THINK HE'S HURT.......
WELL, KILLER, THE DOCTOR SAYS YOU BROKE YOUR
PELVIS
BACK
SKULL
R.ARM
R.LEG
WANT TO JUMP AGAIN?  (Y=YES) Y
HOW MANY BUSSES WILL YOU TRY TO JUMP? 5
 5 BUSSES!  THAT'S  75 FEET!
WHAT RAMP ANGLE WILL YOU USE? 22
HOW FAST WILL YOU LEAVE THE RAMP? 46
      GOOD LUCK!
THERE HE GOES!!!!
**************HE'S SHORT OF THE RAMP ......
I THINK HE'S HURT.......
WELL, KILLER, THE DOCTOR SAYS YOU BROKE YOUR
KNEE
FACE
R.LEG
BACK
NECK
PRIDE
WANT TO JUMP AGAIN?  (Y=YES) Y
HOW MANY BUSSES WILL YOU TRY TO JUMP? 5
 5 BUSSES!  THAT'S  75 FEET!
WHAT RAMP ANGLE WILL YOU USE? 22
HOW FAST WILL YOU LEAVE THE RAMP? 47
      GOOD LUCK!
THERE HE GOES!!!!
***************HE MADE IT !  GREAT JUMP, KILLER!
WANT TO JUMP AGAIN?  (Y=YES) N
YOU MADE IT  1 OUT OF  3 ATTEMPTS.
BE CAREFUL, NOW.
READY
```

Charles Aylworth, 3415 Bell Ave., Eugene, OR 97402.

Program Listing

```
10 REM EVILK MIGHTY MOTORCYCLE JUMP
20 REM PROGRAMMED 12/8/77 BY CHUCK AYLWORTH
30 REM IN NORTHSTAR BASIC.
40 REM I$ IS INJURY STRING(MORE MAY BE ADDED)
50 DIM I2(14),I$(84)
60 REM INITIALIZATION OF RANDOM FUNCTION.
90 I$= "R.ARM L.ARM R.LEG L.LEG BACK  NECK  SKULL RIBS  KNEE  BUTT  "
100 I$=I$+"FACE  PELVISPRIDE BIKE  "
110 T=.1
120 REM T IS RATE OF JUMP CALCULATION.
130 T2=0
140 T3=0
150 PRINT "WE'RE AT THE SCENE OF THE BIG MOTORCYCLE JUMP!"
160 INPUT "HOW MANY BUSSES WILL YOU TRY TO JUMP? ",N
170 J=N*15
180 PRINT N," BUSSES!  THAT'S ",J," FEET!"
190 INPUT "WHAT RAMP ANGLE WILL YOU USE? ",A2
200 IF A2<90 AND A2>0 THEN 230
210 PRINT A2," DEGREES?  THAT'S IMPOSSIBLE.  COME ON NOW, "
220 GOTO 190
230 A=A2*.01745
240 INPUT "HOW FAST WILL YOU LEAVE THE RAMP? ",S
250 PRINT "      GOOD LUCK!"
260 IF S>0 THEN 300
270 PRINT "A PRACTICE JUMP!"
280 PRINT "OK, THIS TIME ",
290 GOTO 240
300 H=6
310 D=0
320 G=6
330 R2=0
340 S2=0
350 S=S*1.5
360 PRINT "THERE HE GOES!!!!"
370 REM JUMP CALCULATION LOOP.
380 S=S-S2
390 F=S*T
400 D2=F *COS(A)
410 R=F*SIN(A)
420 R2=R2+(32*T)
430 R3=R2*T
440 H=H+R-R3
450 D=D+D2
460 PRINT"*",
470 REM S2 IS DRAG FACTOR.
480 S2=(S/120)*32*T
490 IF D>=J THEN G=G-R
500 REM CALCULATE HEIGHT ABOVE GROUND (ALLOW FOR OFFRAMP)
510 IF G<=0 THEN G=0
520 IF H>G THEN 380
530 IF D<J THEN 600
540 IF D>J+20 THEN 640
550 L=((D-J)/30)+RND(0)
560 IF L>.8 THEN 650
570 PRINT "HE MADE IT !  GREAT JUMP, KILLER!"
580 T2=T2+1
590 GOTO 920
600 PRINT "HE'S SHORT OF THE RAMP ......"
610 REM CALCULATION OF INJURIES.
620 L2=INT((((J-D)/5)*2)+(RND(0)*5)+.5)
630 GOTO 670
640 PRINT "HE JUMPED TOO FAR!   ",
650 PRINT "HE MISSED THE RAMP."
660 L2=INT (((D+20-J)/20)+(RND(0)*5))
670 PRINT "I THINK HE'S HURT......."
680 FOR K=1 TO 14
690 I2(K)=K
700 NEXT K
710 K2=14
720 IF L2>14 THEN L2=14
730 IF L2<=0 THEN L2=1
740 REM LISTING OF INJURIES.
750 REM *** RANDOM PERMUTATION OF LENGTH L2.
760 FOR K=1 TO L2
770 V=INT(RND(0)*1000)
780 V=(V-(INT(V/K2)*K2))+1
790 H2=I2(V)
800 I2(V)=I2(K2)
810 I2(K2)=H2
820 K2=K2-1
830 NEXT K
840 PRINT "WELL, KILLER, THE DOCTOR SAYS YOU BROKE YOUR "
850 REM GET SPECIFIC INJURIES FROM LIST.
860 FOR K=(15-L2) TO 14
870 REM SOME BASICS USE MID(I$,P,6).
880 P=(6*I2(K))-5
890 A$=I$(P,P+5)
900 PRINT A$
910 NEXT K
920 T3=T3+1
930 INPUT "WANT TO JUMP AGAIN?  (Y=YES) ",A$
940 IF A$="Y" THEN 160
950 PRINT "YOU MADE IT ",T2," OUT OF ",T3," ATTEMPTS."
960 PRINT "BE CAREFUL, NOW."
970 END
```

Prime Factoring with Quasi-Primes

Jay M. Jeffery

A program that yields the prime factors of integers comes in handy at various levels of mathematics education. It can be used to reduce fractions to lowest terms, to find greatest common factors, to find least common multiples, and to test if numbers are prime.

The following program in BASIC is operated very rapidly even with relatively large numbers. The feature that helps to increase the efficiency is the use of equations that generate all primes consecutively while eliminating many, though not all, composites. The set of numbers generated could be called quasi-primes. Using such a set as a source of divisors in a factoring algorithm reduces the number of unnecessary divisions considerably, compared to programs that employ consecutive integers or the set of odd integers,

Jay M. Jeffery
Mathematics Dept.
Hawken School
Box 249, County Line Road
Gates Mills, Ohio 44040

```
100 PRINT "PRIME FACTORING PROGRAM"
110 PRINT "INPUT POSITIVE INTEGER TO BE FACTORED:"
120 INPUT X1
130 PRINT "THE PRIME FACTORS ARE AS FOLLOWS:"
140 REM:   X2 REPRESENTS THE DIVISORS USED TO TEST NUMBER.
150 X2=2
160 GOSUB 300
170 X2=3
180 GOSUB 300
190 REM:  LINES 170 THROUGH 230 GENERATE ALL PRIMES REQUIRED
200 REM:  AS DIVISORS.  ALSO, SOME COMPOSITES ARE GENERATED,
210 REM:  BUT THE EQUATIONS REDUCE THE NUMBER CONSIDERABLY.
220 FOR N=0 TO SQR(X1)/6
230 X2=6*N+5
240 GOSUB 300
250 X2=6*N+7
260 GOSUB 300
270 NEXT N
280 GOTO 380
290 REM:  SUBROUTINE TESTS DIVISIBILITY.
300 X4=INT(X1/X2)*X2
310 X4=INT(X1/X2)*X2
320 IF X4#X1 THEN 370
330 PRINT X2;
340 X1=X1/X2
350 IF X1=1 THEN 390
360 GOTO 310
370 RETURN
380 PRINT X1
390 END


RUN
PRIFAC

PRIME FACTORINGROGRAM
INPUT POSITIVE INTEGER TO BE FACTORED:
?56
THE PRIME FACTORS ARE AS FOLLOWS:
 2    2    2    7
DONE


RUN
PRIFAC

PRIME FACTORING PROGRAM
INPUT POSITIVE INTEGER TO BE FACTORED:
?9819
THE PRIME FACTORS ARE AS FOLLOWS:
 3    3    1091

DONE
```

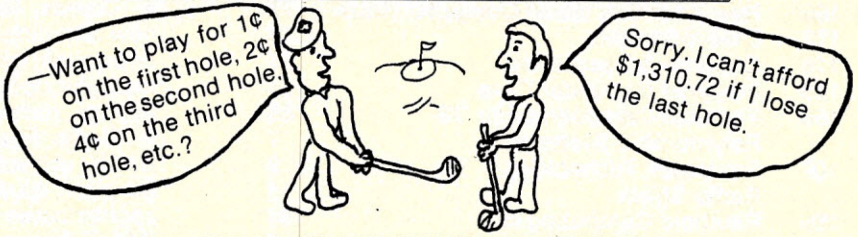
Powers of 2

The Basic computer program below displays the numerical value of

$2^0, 2^1, 2^2, 2^3, \ldots, 2^{17}$

in the second column.

```
10 LET X=0
20 PRINT X,2^X
30 LET X=X+1
40 IF X=18 THEN 60
50 GO TO 20
60 END

READY

RUNNH
 0          1
 1          2
 2          4
 3          8
 4          16
 5          32
 6          64
 7          128
 8          256
 9          512
 10         1024
 11         2048
 12         4096
 13         8192
 14         16384
 15         32768
 16         65536
 17         131072
```



Index to Advertisers

MORE Information MORE

At the right price to you, the reader (namely a 15¢ stamp), *Creative Computing* is providing a new service to help you get all the computer product and service information for which you used to have to spend laborious time writing to each company, finding misplaced stamps, and looking up addresses. No more! You need only circle the numbers on the reader service card to the right that correspond to the companies from which you'd like more information, find *one* stamp, drop it in the mail, and wait for your postman. *Creative Computing* does the rest.

We hope this new service helps us too. Increasing numbers of subscribers are writing *Creative Computing* asking us for information on new products and services they read about each issue. It's impossible for us to stock manufacturer literature, especially in the microcomputer field where state-of-the-art development is so rapid. Instead, we're making it easy for you to get material directly from the sources involved.

Creative Computing is pleased to bring you this speedy, economical way to get information and hopes you feel it is a worthwhile addition to everybody's favorite magazine.

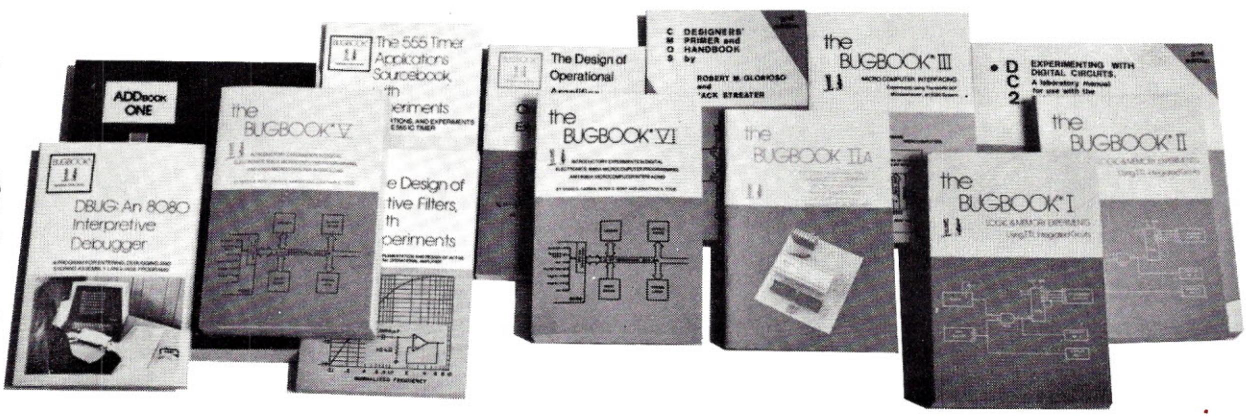
Coming in September

- **Special Section on Simulation and Gaming in Education and Business.** Using simulation and gaming, you can create situations that are often difficult, expensive or impossible to experience any other way. This special group of articles looks into some of the latest techniques and available software in a field that continues to grow at a fast pace.
- **NEPHIS Indexing System.** NEPHIS is a computer-assisted system that can produce sophisticated-looking subject indexes to just about anything, from your personal collection of paperbacks to all the books in the Library of Congress. Best of all, it's easy to implement and to use, even if you know very little about computers and not very much about indexing.
- **Business Computing: Accounts Receivable.** One of the most important components of any accounting system is accounts receivable. The fourth part of this series on business computing examines the offerings of several suppliers of A/R software packages, plus an independent overview of this facet of accounting, with comments on Customer Account Files, Transaction Activity Files, and reports such as Periodic Activity and Invoice Register.
- **Games, Games, Games.** As the leader in publishing new and different computer games, *Creative Computing* once again gives you a couple you'll want to put into your machine right away. And as usual, we provide complete listings, runs and descriptions.
- **Radio Shack's Level-II BASIC.** The TRS-80 Level-I BASIC will suit the needs of many people who will run only simpler games or programs that don't require sophisticated arithmetic functions in ROM. Level-II BASIC has so many more goodies that it uses a 12K ROM (Level-I uses a 4K ROM), and includes print formatting, up to 16 digits of accuracy, editing capabilities, user control of program errors, and trig, log and exponentiation functions. We'll look at the differences between Level-I and Level-II and comment on the just-published Level-II manual.
- **More from ROM.** As in this issue, we'll be printing a special 32-page section of previously unpublished material from ROM magazine, which was recently absorbed by the Creative Computing empire.

Go Bugs Between the Covers.

Between the covers of the Bugbook® Library, you'll find the most comprehensive and authoritative tutorials and reference works in electronics today. Written for both hobbyist and professional, 23 detailed, illustrated volumes carry you through the training ground of basic electronics, starting at the most elementary level all the way to sophisticated techniques with linear circuitry and the 8080A Microprocessor. Learn fundamental circuit designing by implementing computer controls of instrumentation. These texts, manuals and reference series have already become indispensable to over 200,000 buyers.

Uncover the world of electronics. Send for our free Bugworks® catalog with all of the Bugbooks described—the first and last words in electronics today.



☐ **Please send me more information and specific descriptions of each book in your library.**

Name ________________________

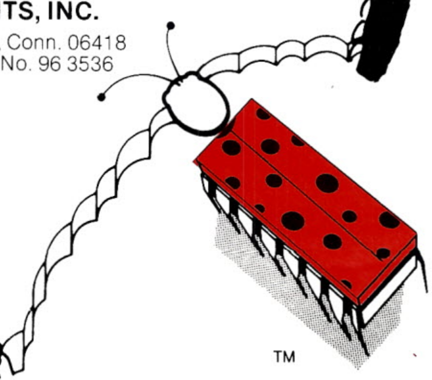
Address ________________________

City ________________________

State ____________ Zip ________

Phone ________________________

E&L INSTRUMENTS, INC.
61 First Street, Derby, Conn. 06418
(203) 735-8774 Telex No. 96 3536



TM

CIRCLE 114 ON READER SERVICE CARD